COPPE/UFRJ

PROCESSAMENTO DE ALTO DESEMPENHO EM CONSULTAS ANALÍTICAS SOBRE BASE DE DADOS GEOESTATÍSTICOS

Melissa Paes Campos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador(es): Marta Lima de Queirós Mattoso

Alexandre de Assis Bento Lima

Rio de Janeiro

Setembro de 2008

PROCESSAMENTO DE ALTO DESEMPENHO EM CONSULTAS ANALÍTICAS SOBRE BASE DE DADOS GEOESTATÍSTICOS

Melissa Paes Campos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Profª. Marta Lima de Queirós Mattoso, D. Sc.

Prof. Alexandre de Assis Bento Lima, D. Sc.

Prof. Geraldo Zimbrão da Silva, D. Sc.

Prof. Luiz Antônio Vivacqua Corrêa Meyer, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2008

Campos, Melissa Paes

Processamento de Alto Desempenho em Consultas Analíticas sobre Base de Dados Geoestatísticos/ Melissa Paes Campos. – Rio de Janeiro: UFRJ/COPPE, 2008.

XII, 114 p.: il.; 29,7 cm.

Orientadores: Marta Lima de Queirós Mattoso

Alexandre de Assis Bento Lima

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2008.

Referencias Bibliográficas: p. 112-114.

1. Agrupamento de Banco de Dados. 2. Projeto de Banco de Dados Distribuídos. 3. Balanceamento Dinâmico de Carga. I. Mattoso, Marta Lima de Queirós *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Titulo.

À minha filha Anabela.

O cérebro humano possui aproximadamente cem bilhões de neurônios. Estimando em cerca de mil as conexões de cada neurônio com seus vizinhos, temos cerca de cem trilhões de conexões, cada qual capaz de um cálculo simultâneo. Isto é um processamento paralelo bastante maciço e uma das chaves para a força do pensamento humano. Entretanto, é uma fraqueza profunda a velocidade aflitivamente lenta dos circuitos neurais, somente de duzentos cálculos por segundo. Para os problemas que se beneficiam do paralelismo maciço, como um reconhecimento de padrões com base na rede neural, o cérebro humano se sai muito bem. Para os problemas que exigem raciocínio sequencial em grande escala, o cérebro humano é apenas medíocre.

(A Era das Máquinas Espirituais, Ray Kurzweil, 1999, p.147)

AGRADECIMENTOS

Ao Instituto Brasileiro de Geografia e Estatística - IBGE, pela concessão de afastamento integral, com ônus, das minhas atividades na Instituição, durante os dois anos de curso, e por ter cedido a base de dados utilizada neste trabalho. A realização do curso de Mestrado nessas condições é uma oportunidade ímpar que uma Instituição pode oferecer ao seu funcionário.

À Professora Marta Mattoso, pela orientação oportuna, confiança e apoio técnico; por incentivar-me em participar de congressos e eventos; e principalmente, pelo esforço em viabilizar a minha participação no evento internacional VECPAR 2008, no qual apresentei e publiquei um artigo, recebendo o prêmio *Best Student Paper Award*.

Ao Alexandre Assis, pela orientação oportuna, confiança e apoio técnico; e por incentivar-me em participar de congressos e eventos.

À minha querida filha Anabela, que apesar dos seus sete anos, demonstrou muita maturidade ao entender a importância desse momento em minha vida, sendo carinhosa, obediente, e principalmente, paciente nas horas em que eu não pude brincar e passear com ela; e que na sua inocência, até ofereceu ajuda para a realização dos experimentos.

À minha mãe Eloiza, pelo seu amor, por sua ajuda nos cuidados de Anabela; por ter me ouvido falar todo o tempo, mesmo sem entender nada (obrigada principalmente, pela expressão de que o assunto era interessante); e pela paciência durante os meus momentos de tensão, preocupação, cansaço e mau humor. E ao Tyrone, seu companheiro e meu padrasto, por também ter ajudado nos cuidados da minha filha e brincado muito com ela.

Ao meu amado Leo, por ter estado presente em todas as etapas do desenvolvimento desta dissertação até o final, sempre com seu bom humor, alegria, paciência, carinho e muito amor; por seu interesse e compreensão técnica pelo tema da pesquisa, nos permitindo conversar a respeito durante nossos passeios e jantares, donde surgiram ótimas idéias e melhorias que foram aplicadas a este trabalho; e principalmente por ele ter aceitado me ouvir falando de trabalho quase todo o tempo em que estávamos juntos, trocando as palavras de amor e carinho por alto desempenho e OLAP.

Ao meu amigo e colega de trabalho José Sant'Anna Bevilaqua, pela total confiança em meu trabalho, desde o meu ingresso no IBGE, e pela motivação para a realização deste curso de Mestrado; por ter acreditado que eu seria capaz de realizar esta pesquisa; pelas sugestões dadas ao uso da base de dados e das consultas dos experimentos; e principalmente, pela sua amizade.

À minha amiga Patrícia Barros, companheira de curso, por ter me ajudado a manter a calma e me incentivado nos momentos difíceis, oferecendo seu ombro e sua atenção, mesmo quando ela estava ocupada com o próprio Mestrado.

Aos meus amigos da Equipe BME, por terem me apoiado durante o meu retorno à Instituição e na finalização da minha dissertação. São eles: Antônio Fernando, Bianca, Magali e Sandra.

Ao Professor Patrick Valduriez, por ter fornecido os recursos computacionais no INRIA, França, para a realização dos meus experimentos. À equipe de suporte do Grid5000, em especial, Pascal Morillon, sempre solícito às minhas dúvidas em relação ao uso do Cluster do Grupo Paris de IRISA, Rennes.

Agradeço a Deus por essa realização.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROCESSAMENTO DE ALTO DESEMPENHO EM CONSULTAS ANALÍTICAS SOBRE BASE DE DADOS GEOESTATÍSTICOS

Melissa Paes Campos

Setembro/2008

Orientadores: Marta Lima de Queirós Mattoso
Alexandre de Assis Bento Lima

Programa: Engenharia de Sistemas e Computação

As consultas analíticas apresentam alto custo de processamento e podem ser de longa duração, pois realizam operações complexas sobre massas de dados de tamanho significativo. O tempo gasto na obtenção de uma informação é imperativo para o processo de tomada de decisão. O ParGRES é uma solução de código aberto, desenvolvida para ser uma camada intermediária entre o banco de dados e uma aplicação cliente em um agrupamento de banco de dados, provendo paralelismo intra-consulta no processamento de consultas. Através de experimentos utilizando a base de dados sintética do *benchmark* TPC-H, o ParGRES apresentou grande eficiência no desempenho de processamento de consultas, motivando a sua avaliação com uma base de dados real. Utilizou-se uma base de dados geoestatísticos produzida pelo IBGE, na qual são realizadas consultas analíticas específicas e complexas. Os experimentos foram realizados em base total e parcialmente replicada. Excelentes resultados foram obtidos no que tange a redução do tempo de processamento das consultas, as quais obtiveram, na maioria das vezes, aceleração super-linear em todas as configurações do agrupamento. Os resultados obtidos mostram que o ParGRES é uma boa alternativa, com baixo custo de implementação, para aumento de desempenho no processamento de consultas analíticas em cenários reais, tanto com bases totalmente replicadas quanto com bases parcialmente replicadas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

HIGH-PERFORMANCE ANALYTICAL QUERY PROCESSING ON A GEOSTATISTICAL DATABASE

Melissa Paes Campos

September/2008

Advisors: Marta Lima de Queirós Mattoso
Alexandre de Assis Bento Lima

Department: Computer Science and Systems Engineering

Analytical queries typically have high processing costs and can take long time to be processed, due to the complex performed on huge amounts of data. So, speeding up the execution of each single query is imperative to decision making. ParGRES is an open-source database cluster middleware for high performance OLAP query processing, exploiting inter and intra-query parallelism. Through experiments using the synthetic database of the TPC-H benchmark, ParGRES showed excellent performance during query processing, motivating its evaluation on a real-world OLAP database. The Geoestatistic Database, in which complex analytical queries are processed, is provided by IBGE. Experiments were performed using fully and partially replicated databases, and excellent results were obtained, yielding linear and very often super-linear speedup for different experiment setups. The results obtained show that ParGRES is a very cost-effective solution for OLAP query processing in real settings with fully or partially replicated databases.

ÍNDICE

LISTA DE SIGLAS

BME – Banco Multidimensional de Estatísticas

IBGE – Instituto Brasileiro de Geografia e Estatística

OLAP – Online Analytical Processing

OLTP – Online Transactional Processing

SGBD – Sistema Gerenciador de Bancos de Dados

SQL – *Structured Query Language*

CD2000 – Censo Demográfico 2000

AVP – Fragmentação Virtual Adaptativa (*Virtual Partitioning Adaptive*)

BD – Banco de Dados

DW – *Data Warehouse*

BDD – Banco de Dados Distribuído

FV – Fragmentação Virtual

AFV – Atributo de Fragmentação Virtual

FH – Fragmentação Híbrida

CQP – Processador de Consultas do Cluster (*Cluster Query Processor*)

GQT – Tarefa de Consulta Global (*Global Query Task*)

GRC – Coletor Global de Resultados (*Global Result Collector*)

NQP – Processador de Consultas de Nó (*Node Query Processor*)

LQT – Tarefa de Consulta Local (*Local Query Task*)

QE – Executor de Consultas (Query Executor)

LRC – Coletor Local de Resultados (*Local Result Collector*)

1 Introdução

Dentre as diversas atividades realizadas em uma organização (seja esta acadêmica, científica, comercial, empresarial, dentre outras), o processamento das informações é uma atividade fundamental para a realização das outras atividades. Segundo THOMSEN [39], o processamento de informações envolve a coleta, o armazenamento, o transporte, a manipulação e a recuperação de dados (com ou sem a ajuda de computadores). Assim, a informação é o resultado da organização e do relacionamento significativo entre os dados.

Uma vez gerada, a informação pode e deve ser utilizada por pessoas, principalmente as que desempenham tarefas relacionadas à tomada de decisão, à análise de uma tendência ou à previsão da ocorrência de um fato, como os analistas, administradores, gerentes e pesquisadores.

A Tecnologia da Informação vem sendo amplamente utilizada no processamento de informações, principalmente no suporte às tarefas de planejamento e no processo de tomada de decisões. Existem diversas tecnologias que ajudam os analistas e gerentes em suas atividades relacionadas à tomada de decisão. Armazém de Dados[1] (*Data Warehouse* – DW) e Processamento Analítico em Tempo Real (*Online Analytical Processing* - OLAP) são tecnologias que dão suporte a essas atividades, oferecendo a possibilidade de armazenamento de grandes conjuntos de dados e a disponibilização de ferramentas voltadas à sua recuperação e análise.

Armazém de dados é um banco de dados especializado no armazenamento, gerenciamento e integração de dados oriundos de fontes diferentes, organizados por assunto, espaço e tempo [18]. A organização desse banco de dados permite ao usuário visualizar uma parte ou todo o negócio da empresa através de uma perspectiva multidimensional, com o objetivo de dar suporte às atividades de análise dos dados.

Ferramentas OLAP são muito utilizadas para a análise de dados em um armazém de dados, permitindo a sua exploração pelo analista ou gerente que, ao conhecer a questão, formula a pergunta e realiza a consulta sobre os dados para obter a resposta. Sua abordagem multidimensional de análise oferece ao usuário a possibilidade de realização de consultas de forma dimensional, natural e intuitiva [7]. As consultas OLAP são complexas, podem ser de longa duração e apresentam alto custo de

processamento porque envolvem acesso a diferentes níveis de granularidade dos dados, realizam diversos agrupamentos, agregações e cruzamentos de dados. Ou seja, o processamento de uma consulta desse tipo pode levar horas ou dias. O tempo gasto na obtenção de uma informação é imperativo ao processo de tomada de decisão, devido ao cenário dinâmico e competitivo de um analista envolvido com as atividades deste processo.

Atualmente, um dos maiores desafios em relação às consultas OLAP é a redução de seu custo relacionado ao tempo de processamento. VALDURIEZ [43] propõe o uso de sistemas de Bancos de Dados Paralelos como solução para o aumento de desempenho no processamento de consultas. Os Bancos de Dados Paralelos são sistemas que combinam multiprocessadores fortemente acoplados e distribuição (fragmentação e replicação) de dados para obter processamento paralelo, explorando o paralelismo inter-consulta e intra-consulta. Através do paralelismo inter-consulta, várias consultas podem ser executadas ao mesmo tempo (em paralelo), enquanto que no paralelismo intra-consulta, uma consulta é dividida e distribuída para ser executada por vários computadores. Por processar cada consulta sequencialmente, mas em paralelo, o paralelismo inter-consulta aumenta a vazão do processamento de transações. Esse tipo de paralelismo privilegia o processamento de consultas OLTP (*Online Transaction Processing*, ou Processamento de Transações em Tempo Real), ao executar o maior número de consultas possíveis em paralelo. Entretanto, para o processamento de consultas OLAP, o paralelismo intra-consulta é essencial, pois além de reduzir o tempo de processamento individual de cada consulta, reduzirá o tempo total de processamento de todas as consultas.

Apesar de eficiente, o uso de Bancos de Dados Paralelos é uma solução cara, na qual estão envolvidos custos com programas proprietários, equipamentos e projeto físico de banco de dados.

A abordagem de agrupamento de banco de dados é uma forma de obter alto desempenho no processamento de consultas através de paralelização com baixo custo de implementação em um agrupamento de computadores[2] [1, 6, 22, 24]. AKAL *et al.* [1] define agrupamento de banco de dados como sendo um conjunto de computadores (nós) que se comunicam através de uma rede, no qual cada computador possui uma instância

[1] Armazém de Dados, conhecido na literatura como *Data Warehouse* – DW.
[2] Agrupamento de Computadores, conhecido na literatura como *PC Cluster* ou *Cluster*

de sistema de gerência de banco de dados (SGBD) sequencial e não paralelo, e um aplicativo central para gerenciar esse conjunto de SGBD.

Seguindo essa abordagem, o ParGRES [24, 29] foi desenvolvido para ser uma camada intermediária de *software* (de código aberto) entre um banco de dados e uma aplicação cliente em um agrupamento de banco de dados, provendo paralelismo inter e intra-consulta no processamento de consultas OLAP. Por ser uma camada intermediária entre a aplicação cliente e um banco de dados, o ParGRES intercepta cada consulta e coordena a sua execução com a ajuda dos bancos de dados sequenciais instalados em cada nó do agrupamento. O ParGRES utiliza uma variação da fragmentação virtual, denominada Fragmentação Virtual Adaptativa [21], e replicação total da base de dados para realizar balanceamento de carga durante o processamento paralelo intra-consulta.

Aliando as técnicas de processamento paralelo de consultas ao agrupamento de banco de dados, algumas soluções foram propostas para aumentar o desempenho de consultas OLAP através de paralelização, como PowerDB [31], C-JDBC [6], Apuama [25] e Sequoia [35]. Dentre essas soluções, apenas o PowerDB e o ParGRES oferecem paralelismo intra-consulta, mas o PowerDB apresenta diversas limitações em relação ao balanceamento de carga durante o processamento de consultas. A eficiência do ParGRES foi validada através de experimentos utilizando o *benchmark* TPC-H [41]. Este *benchmark* representa aplicações OLAP e possui vinte e duas consultas de alto custo e duas atualizações. Bons resultados foram obtidos em MATTOSO *et al.* [24] no que tange a redução do tempo de processamento das consultas, motivando a sua avaliação no processamento de consultas analíticas em uma base de dados real. Utilizamos uma base de dados geoestatísticos produzida pelo Instituto Brasileiro de Geografia e Estatística - IBGE, na qual são realizadas consultas analíticas específicas e complexas.

O IBGE [16] é o principal provedor de dados e informações sobre o Brasil, atendendo às necessidades dos mais diversos segmentos da sociedade civil, bem como dos órgãos das esferas governamentais federal, estadual e municipal. A Internet é o principal canal de comunicação entre o IBGE e o usuário, onde se encontra disponível um dos maiores acervos especializados em informações estatísticas e geográficas do país. O Banco Multidimensional de Estatísticas – BME [3], um armazém de dados desenvolvido pela Instituição e disponível na Internet, que tem como objetivo disponibilizar ao público ferramentas voltadas à busca, recuperação e manuseio das informações estatísticas, de forma totalmente desagregada e multidimensional. O Censo

Demográfico 2000 [4], pesquisa realizada pelo IBGE e fonte de informações sobre a situação de vida da população do Brasil, é uma das pesquisas disponíveis no BME e possui cerca de 250 milhões de registros.

De acordo com o contexto descrito, temos um cenário típico de Sistemas de Informações Multidimensionais de Apoio a Decisão, com grande volume de dados e consultas analíticas complexas e de alto custo. Assim, o Censo Demográfico 2000 e o BME são os elementos essenciais, junto com a tecnologia de agrupamento de banco de dados e processamento paralelo, para a criação de um ambiente real e ideal para a avaliação do ParGRES no processamento de consultas OLAP utilizando paralelismo intra-consulta. Portanto, a base de dados e as consultas OLAP utilizadas nos experimentos dessa dissertação são oriundas do Censo Demográfico 2000 e do BME, respectivamente.

O objetivo desta dissertação é avaliar o desempenho do ParGRES no processamento de consultas OLAP em uma base de dados real do Instituto Brasileiro de Geografia e Estatísticas – IBGE, e descrever uma metodologia para transformar uma aplicação baseada em consultas sequenciais OLAP em uma aplicação que possibilite a realização de consultas paralelas utilizando o ParGRES.

O foco dos nossos experimentos é analisar a abordagem não intrusiva da Fragmentação Virtual Adaptativa no paralelismo intra-consulta e no balanceamento de carga durante o processamento de consultas. Para a realização dos experimentos, utilizamos a base de dados da Amostra do Censo Demográfico 2000, composta por três tabelas de fatos, totalizando aproximadamente 30 milhões de tuplas, e quatorze consultas *Ad-hoc* oriundas do BME. Essas consultas possuem diferentes níveis de complexidade, incluindo diversas junções entre tabelas de fatos e de dimensões, muitas agregações e predicados de seleção com variados fatores de seletividade.

Para utilizar o ParGRES com a base do Censo Demográfico 2000, foi necessário apenas informar os nomes das tabelas de fatos e seus atributos de fragmentação virtual, e criar um índice de agrupamento sobre esses atributos. Todos os experimentos foram executados em um agrupamento de banco de dados de 64 nós do projeto Grid5000 [11], situado em Rennes, na França. Este projeto tem como objetivo disponibilizar uma infra-estrutura de larga escala (Grade[3]) que pode ser reconfigurada, controlada e monitorada para a realização de experimentos acadêmicos de processamento paralelo e distribuído.

[3] Grade, conhecido na literatura como *Grid*.

Utilizamos o PostgreSQL 8.2.4 [30] para gerenciar a base de dados do experimento, distribuída entre os nós do agrupamento.

Os experimentos foram divididos em duas partes: na primeira, as consultas foram executadas isoladamente com diferentes números de nós do agrupamento, e, na segunda, as consultas foram organizadas em lotes e executadas concorrentemente, também com diferentes números de nós. Ambos os experimentos foram realizados com base total e parcialmente replicada entre os nós. Nos experimentos realizados com a base totalmente replicada, com consultas isoladas, foram obtidos resultados excelentes (aceleração super-linear na maioria dos casos), com todas as consultas apresentando redução no tempo de processamento à medida que aumentamos o número de nós do agrupamento, conforme apresentado em PAES [28]. Os resultados foram similares nos experimentos com concorrência.

Apesar dos bons resultados obtidos nesses experimentos, o uso de uma base totalmente replicada em todos os nós do agrupamento pode dificultar a utilização do ParGRES como solução para aumento de desempenho no processamento de consultas OLAP, pois os armazéns de dados tendem a armazenar um grande volume de dados. FURTADO [10] desenvolveu um protótipo, denominado SmaQSS, que une a Fragmentação Virtual Adaptativa e a fragmentação híbrida do banco de dados para prover paralelismo intra-consulta em bases de dados parcialmente replicadas. Seguindo a abordagem deste protótipo, nós fizemos algumas modificações no ParGRES, para o mesmo prover paralelismo intra-consulta em bases parcialmente replicadas. Essas modificações incluem o algoritmo de difusão de mensagens de oferta de ajuda e o projeto de distribuição. Novos experimentos foram realizados sobre o ParGRES (modificado) para avaliar o desempenho do processamento de consultas em uma base parcialmente replicada. Também alcançamos bons resultados nesses experimentos, entretanto, houve um leve aumento dos tempos de processamento das consultas (isoladas e concorrentes) quando comparado com os tempos dos experimentos com base totalmente replicada. Apesar desse aumento nos tempos, os resultados mostram que o ParGRES pode ser uma solução com baixo custo de implementação para aumento de desempenho no processamento de consultas OLAP em cenários reais, tanto com bases totalmente replicadas quanto com bases parcialmente replicadas.

Esta dissertação está organizada da seguinte maneira. No capítulo 2, discutimos os conceitos relacionados à OLAP e ao processamento paralelo, e os trabalhos correlatos ao desta dissertação. No capítulo 3, descrevemos uma metodologia para uso do

ParGRES em bases reais e as modificações realizadas para o seu funcionamento com bases parcialmente replicadas. No capítulo 4, apresentamos o Censo Demográfico 2000 e o Banco Multidimensional de Estatísticas, de onde foram extraídas a base de dados e as consultas OLAP utilizadas nos experimentos desta dissertação. No capítulo 5, descrevemos os experimentos realizados e apresentamos os resultados obtidos. O capítulo 6 conclui o trabalho.

2 Processamento paralelo de consultas OLAP

Esta dissertação se baseia nos conceitos relacionados à OLAP e processamento paralelo. Neste capítulo é feita uma revisão sobre os assuntos afins para proporcionar ao leitor uma melhor compreensão das atividades desenvolvidas ao longo deste trabalho. Na seção 2.1 apresentamos o termo OLAP, a base de dados e as consultas de análise as quais essa tecnologia dá suporte; na seção 2.2 descrevemos o projeto de distribuição de uma base de dados, a fragmentação de tabelas e a alocação dos fragmentos gerados; na seção 2.3 descrevemos o processamento paralelo e como as consultas OLAP são processadas utilizando paralelização; na seção 2.4 apresentamos o conceito de fragmentação virtual; nas seções 2.5 e 2.6 apresentamos o ParGRES e o SmaQSS, as técnicas de fragmentação virtual adaptativa e híbrida, e as suas arquiteturas; e por fim, na seção 2.7, analisamos os trabalhos similares ao realizado nesta dissertação: o PowerDB, C-JDBC, Apuama e Sequoia.

2.1 Consultas OLAP

Em geral, os dados de uma organização são armazenados em uma estrutura regular digital, denominada Banco de Dados - BD, que nada mais é do que uma coleção de dados organizada dentro de um computador. Esses dados são gerenciados por um sistema específico, denominado Sistema Gerenciador de Banco de Dados – SGBD, onde são realizados diversos procedimentos, ou transações do tipo OLTP, que permitem a inserção, a alteração, a exclusão e o acesso aos dados. OLTP é uma sigla em inglês para *Online Transaction Processing*, ou Processamento de Transações *On-line*, que são transações que registram as diversas operações existentes em uma determinada atividade organizacional. Controle de estoque de produtos e movimentação financeira de uma empresa são exemplos de transações operacionais.

Em uma empresa, independente do tipo de negócio, são realizadas atividades operacionais e de análise. As atividades operacionais produzem dados relacionados diretamente com o dia a dia da empresa, ou seja, dados gerados pelo seu negócio. As transações OLTP dão suporte a essas atividades no que tange o armazenamento desses dados em um BD. As atividades de análise manuseiam os dados produzidos para obterem informações a respeito do negócio e permitir a tomada de decisão.

Com o advento da necessidade de analisar os dados gerados ao longo de um determinado período e assim conduzir uma organização, surgiu uma nova abordagem de armazenamento e acesso aos dados de uma empresa, o Armazém de Dados.

Segundo ELMASRI e NAVATHE [9], armazém de dados é uma coleção de dados (ou bancos de dados), oriundos de fontes diferentes, organizados por assunto, espaço e tempo, que fornecem armazenamento, funcionalidade e capacidade de responder consultas de análise complexa, descoberta de conhecimento e tomada de decisão. Os dados de um armazém de dados são provenientes de sistemas transacionais OLTP, que juntos formam séries históricas possibilitando a previsão de eventos futuros com base na análise de eventos passados.

O armazenamento e a visualização de dados de um armazém de dados possuem uma perspectiva multidimensional. Essa perspectiva pode ser ilustrada através de um cubo, cuja estrutura tem formato multidimensional. Esse cubo é formado por três ou mais dimensões, que são unidades de análise dos dados, e se referem ao espaço, ao tempo e ao tipo de dado armazenado; e por medidas, que são valores que representam um dado. Algumas dimensões possuem uma hierarquia entre si para representarem a granularidade dos dados armazenados. Cada célula do cubo representa um dado. A Figura 1 ilustra um cubo de um armazém de dados.

Os armazéns de dados são projetados utilizando a modelagem multidimensional como técnica de projeto lógico, para obter uma representação concreta do cubo acima ilustrado. O modelo multidimensional gerado é composto por três elementos essenciais: o fato, a medida e a dimensão. Segundo MACHADO [23], o fato é uma coleção de itens de dados de medidas e de contexto; a medida é um atributo numérico que representa o fato; e a dimensão fornece informação de contexto ao fato analisado. Para a representação gráfica desses elementos, KIMBALL *et al.* [18] propôs dois modelos dimensionais: Estrela (*Star*) e Floco de Neve (*Snow Flake*). O modelo Estrela é composto por uma tabela de fatos (dominante, no centro do modelo), que contém medidas e valores de negócio, ligadas (através de junções) às tabelas de dimensões, que contêm descrições textuais das dimensões do negócio. Se um modelo Estrela possui mais de uma tabela de fato, temos uma constelação de fatos. Uma constelação de fatos pode ser definida como um conjunto de tabelas de fatos que compartilham algumas tabelas de dimensão.

O modelo Floco de Neve é uma variação do modelo Estrela, onde uma tabela de dimensão se divide em várias outras tabelas, objetivando o uso eficiente de espaço em

disco. Porém, esse modelo torna o esquema um pouco mais complexo, pois utiliza mais tabelas para representar as mesmas dimensões e aumenta a dificuldade na navegação das tabelas.

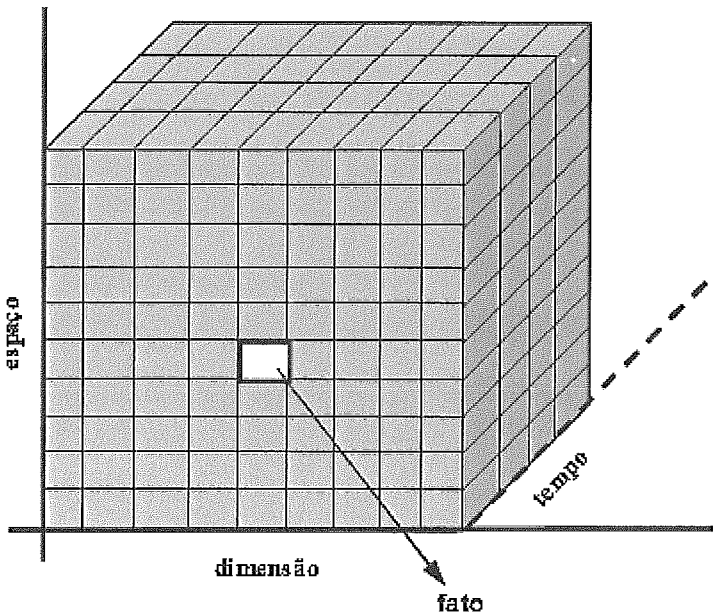


Figura 1: Ilustração da perspectiva em cubo de um armazém de dados

Uma das ferramentas mais utilizadas para a análise de dados em um armazém de dados são as ferramentas de consultas OLAP, que permitem a exploração dos dados com base na verificação, onde o analista ou gerente conhece a questão, formula a pergunta e utiliza uma das ferramentas para realizar a consulta sobre os dados e obter a resposta.

O termo OLAP, utilizado pela primeira vez em 1993 por CODD *et al.* [7], é uma sigla em inglês para *Online Analytical Processing*, ou Processamento Analítico *On-line*, sendo uma variação do termo OLTP, que fornece subsídios a um gerente para realizar consultas analíticas sobre os dados através de uma perspectiva dimensional, natural e intuitiva. As consultas são feitas sobre os diferentes níveis de granularidade dos dados e de diversas formas, realizando agregações e agrupamento de dados, através de filtros e operações existentes, como por exemplo:

- *Slice* e *Dice*: realizam restrições ou não, criando visões dos dados para serem visualizados através de diferentes perspectivas;
- *Drill-Down* e *Roll-Up*: desagrega ou agrega uma dimensão;
- *Drill-through*: recupera os dados originais que geram os dados agregados;

- *Pivot*: muda a dimensão temporal de posição, modificando o eixo de visualização;
- *Rank*: ordena os itens de uma dimensão utilizando um critério.

As consultas OLAP são de natureza não previsível e específica (*Ad-hoc*), definidas de acordo com os interesses do pesquisador e que envolvem um grande volume de dados. Neste sentido, OLAP oferece suporte às atividades de análise.

THOMSEN [39] faz uma comparação interessante entre as atividades operacionais e de análise, que permite apresentar as principais diferenças entre OLTP e OLAP (visto que são esses tipos de transações em BD que dão suporte a essas atividades). A Tabela 1 mostra essa comparação e permite dimensionar mais claramente a complexidade envolvida com as consultas OLAP e o seu significado.

Tabela 1: Comparação entre transações OLTP e OLAP. **Fonte**: THOMSEN [39]

Transações OLTP (suporte às atividades operacionais)	**Transações OLAP (suporte às atividades de análise)**
Mais freqüentes	Menos freqüentes
Mais previsíveis	Menos previsíveis
Menores quantidades de dados acessados por consulta	Maiores quantidades de dados acessados por consulta
Consulta principalmente de dados primitivos	Consulta principalmente de dados derivados
Exige principalmente dados atuais	Exigem dados passados, presentes e projetados
Pouca ou nenhuma derivação complexa	Muitas derivações complexas

2.2 Projeto de distribuição

Com o avanço da descentralização dos dados e das redes de computadores surgiram os BD distribuídos – BDD, que são um conjunto de banco de dados interconectados por uma rede e que realizam tarefas de forma cooperativa. Uma questão importante em um BDD é a disposição física dos dados entre os sítios (computadores). Essa questão é tratada no Projeto de Distribuição de Dados, através do uso de técnicas que dividem as relações (tabelas) de um BD em unidades menores e as distribuem pelos computadores. Essas unidades menores são denominadas fragmentos.

As técnicas envolvidas no projeto de distribuição são a fragmentação, a replicação e a alocação de dados. A fragmentação de dados consiste em definir como as tabelas podem ser divididas e quantos fragmentos serão gerados, levando-se em conta as características envolvidas nas consultas realizadas sobre as bases de dados. Essa técnica será descrita na seção 2.2.1. A replicação de dados consiste em definir o número de

cópias (réplicas) de cada fragmento gerado, a fim de oferecer maior disponibilidade dos dados. A alocação define a localização das réplicas, para otimizar o acesso e também a disponibilidade dos dados. Essas duas últimas técnicas serão apresentadas na seção 2.2.2.

2.2.1 Fragmentação

Uma base de dados pode ser dividida em partes e distribuída em vários computadores, permitindo que várias transações sejam executadas de forma concorrente. A divisão é feita sobre todas as tabelas do banco de dados, segundo uma estratégia de fragmentação a ser escolhida pelo projetista. Antes de escolher a estratégia de fragmentação, é preciso definir o grau de fragmentação das tabelas, que pode variar entre "nenhuma fragmentação" até "fragmentação no nível de tuplas". As estratégias de fragmentação são Fragmentação Horizontal, Fragmentação Vertical e Fragmentação Híbrida. Para utilizar uma dessas estratégias, faz-se necessário ter em mãos informações do banco de dados e informações de aplicativos. As informações do banco de dados são obtidas a partir do esquema conceitual global e dos relacionamentos existentes entre as tabelas. E as informações de aplicativos são obtidas a partir da análise sobre os predicados presentes nas consultas definidas pelos usuários, a seletividade desses predicados (número de linhas retornadas) e a freqüência de realização dessas consultas.

Fragmentação Horizontal. Essa estratégia de fragmentação particiona uma relação horizontalmente, ou seja, produz subconjuntos de linhas através da definição de predicados sobre a relação. Dentre os algoritmos de fragmentação horizontal existentes, ÖSZU e VALDURIEZ [27] apresentam um que realiza o particionamento de uma tabela através da seleção de predicados usados em consultas dos usuários, descrito a seguir: o primeiro passo é agrupar os predicados simples, presentes nas consultas dos usuários; em seguida, identificar os predicados *minterm*, que é a combinação booleana dos predicados simples; no terceiro passo definem-se as implicações entre os predicados simples e os predicados *minterm*; e por último, eliminam-se alguns predicados que são contraditórios de acordo com as implicações definidas. O conjunto de predicados restante após o uso desse algoritmo será utilizado para a seleção das tuplas dos fragmentos a serem gerados de uma dada tabela. A fragmentação horizontal pode ser primária ou derivada. A fragmentação horizontal primária particiona uma tabela baseando-se nos predicados das consultas feitas sobre a tabela proprietária (que por

acaso é a tabela sendo particionada); e a fragmentação horizontal derivada particiona uma tabela membro seguindo a fragmentação realizada sobre a sua tabela proprietária.

Fragmentação Vertical. Essa estratégia de fragmentação particiona uma relação verticalmente, ou seja, produz subconjuntos de atributos de uma relação. Nos fragmentos gerados apenas o atributo de identificação (chave primária) pode estar presente em todos os fragmentos, pois através desse atributo é possível juntar as suas linhas em algum momento. De uma forma muito superficial, podemos dizer que o particionamento da relação é feito verificando a freqüência em que os atributos aparecem nas consultas e agrupando-os segundo uma afinidade, e em seguida, escolhendo os atributos mais utilizados para a definição dos fragmentos. Essa estratégia utiliza dois algoritmos [27]: primeiro, o algoritmo de agrupamento em *clusters*, para agrupar os atributos de uma relação de acordo com a afinidade entre eles; e depois, o algoritmo de particionamento, para encontrar grupos de atributos que são acessados por diferentes aplicativos.

Fragmentação Híbrida. Essa estratégia de fragmentação combina a fragmentação horizontal e vertical quando o uso de uma das estratégias não atende às necessidades dos aplicativos.

Após a fragmentação, o projetista deve aplicar as Regras de Correção de fragmentação para assegurar a consistência do BD, ou seja, garantir que o BD não sofra nenhuma mudança durante a fragmentação e permitir que o mesmo seja reconstruído através de operações de união e junção. As três regras de correção são Completeza, Reconstrução e Disjunção [27]. A regra Completeza assegura que todos os itens de uma relação (tupla ou atributo) estejam presentes em um dos fragmentos gerados. A regra Reconstrução assegura que uma relação possa ser redefinida a partir da união/junção dos seus fragmentos. E a regra Disjunção assegura que um determinado item da relação (tupla ou atributo) esteja presente em apenas um fragmento gerado, com exceção do atributo de identificação (chave primária) no caso de uma fragmentação vertical.

No caso de BD que atendem os sistemas OLAP, torna-se complexo definir um projeto de fragmentação, pois as consultas realizadas na base são, na maioria das vezes, desconhecidas.

2.2.2 Alocação

Após a geração dos diversos fragmentos das relações de um BD, uma questão importante que o projetista do BDD deve se preocupar é em relação à localização desses

fragmentos nos computadores disponíveis, para minimizar o custo com o uso de recursos de computadores e de rede. O custo total do uso desses recursos (de computadores e de rede) é o somatório do custo de armazenamento, de processamento e do tempo de resposta. Para a escolha da localização dos fragmentos, faz-se necessário ter em mãos informações sobre o sítio e informações sobre a rede (além das informações de banco de dados e de aplicativos descritos na seção anterior). As informações do sítio são aquelas a respeito do espaço disponível para o armazenamento e a sua capacidade de processamento. E as informações de rede são aquelas a respeito do custo da comunicação entre os computadores.

Após definir o modelo de alocação dos fragmentos de um BD, o projetista deve decidir pela replicação ou não desses fragmentos. A etapa de replicação consiste em decidir se serão geradas cópias, denominadas réplicas, dos fragmentos e qual o grau de replicação, isto é, o número de réplicas de cada fragmento. A principal vantagem do uso de replicação de uma base de dados é o aumento da disponibilidade dos dados, principalmente durante a recuperação, pois a consulta poderá ser realizada em qualquer computador ou processada em paralelo. No entanto, dependendo do grau de replicação, há perda de desempenho e aumento do custo no que tange às transações de atualização, pois essas deverão ser executadas sobre todas as réplicas a fim de manter a consistência dos dados. O grau de replicação pode variar entre total, nenhuma ou parcial.

Replicação Total. A replicação total consiste em distribuir o banco de dados inteiro em todos os computadores. Esse tipo de replicação garante totalmente a disponibilidade dos dados, bastando apenas um sítio estar no ar. Porém, o custo de atualização da base é altíssimo.

Replicação Parcial. A replicação parcial consiste em distribuir partes do banco de dados entre os computadores. Esse tipo de replicação equilibra as vantagens e desvantagens da replicação total e da não replicação de um BD, oferecendo disponibilidade e minimizando o custo de atualização dos fragmentos.

Se o projetista optar pela não replicação da base de dados, o custo de atualização será mínimo, no entanto, se um computador falhar, não será possível acessar os dados nele contidos.

Uma vez definida a replicação de dados, é necessário definir a localização das réplicas. Existem alguns modelos de alocação de réplicas e três deles foram avaliados e descritos por HSIAO e DEWITT [13]: arquitetura de discos espelhados, particionamento intercalado e particionamento encadeado. A arquitetura de discos

espelhados, implementada pelo sistema Non Stop SQL da Tandem [37], consiste em manter um espelho do disco onde os dados estão armazenados, isto é, manter uma cópia da tabela primária e da réplica em outro disco. A desvantagem do uso desse modelo é que ele não garante a disponibilidade dos dados, pois o espelhamento dos dados é feito no mesmo computador, mas em discos diferentes. O particionamento intercalado do Teradata [38] consiste em manter a tabela primária em um nó, e fragmentar a réplica em **n-1** fragmentos de réplica, onde **n** é o número de nós, e distribuir esses fragmentos de réplicas entre os nós seguintes adjacentes. A desvantagem desse modelo de alocação é a dificuldade em reconstruir a réplica original, além de não garantir a disponibilidade dos dados. Se um nó falhar, não é possível acessar todos os dados. O particionamento encadeado [13] consiste em manter a tabela primária em um nó e alocar a réplica dessa tabela no nó seguinte adjacente. Se a tabela possuir 2 réplicas, essas são armazenadas nos dois nós seguintes adjacentes. E assim por diante, de acordo com o número de réplicas. Esse modelo provê uma maior disponibilidade dos dados.

No caso de BD que atendem os sistemas OLAP, as transações de atualização de dados ocorrem eventualmente. A replicação de dados nesse contexto apresenta mais vantagens do que desvantagens.

2.3 Processamento paralelo de consultas OLAP

Uma das principais demandas sobre uma base de dados é o acesso aos dados nela contidos, o qual é feito através de consultas. As consultas são definidas pelo usuário através de uma linguagem não procedural e de alto nível, por exemplo, o SQL – *Structured Query Language*, e enviadas ao SGBD, responsável em transformar a consulta em uma linguagem que pode ser entendida e processada pelo computador. O Processador de Consultas é um módulo do SGBD responsável por operações de processamento, otimização e execução de uma consulta. Esse conjunto de operações denomina-se Processamento de consultas e é composto por quatro fases: Tradução e Validação, Decomposição, Otimização e Execução de consultas.

Na fase de Tradução e Validação, a consulta é validada através da análise léxica e sintática, que identifica os itens da linguagem e a sintaxe desses itens, e depois traduzida para um código próprio de banco de dados (baseado em álgebra relacional) para ser executada nas fases seguintes. Na fase de Decomposição, a consulta é reescrita para eliminar predicados redundantes, para simplificar as expressões, para separar as

sub-consultas presentes em uma consulta e tratar a localização dos dados envolvidos na consulta. Um ou mais códigos podem ser gerados, denominados planos de execução. Na fase de Otimização decide-se que índices e métodos serão utilizados, em que ordem as operações serão executadas, e escolhe-se o melhor plano de execução (dentre os planos gerados na fase anterior), ou seja, aquele que apresenta o menor consumo de recursos do computador. A fase de Execução transforma o plano escolhido em um código executável e o envia para processamento.

O processamento de consultas é um aspecto de grande importância na implementação de um SGBD, pois envolve uma questão fundamental no uso de um BD: o tempo de resposta de uma consulta. Quanto maior o tempo de resposta, pior é o desempenho do processamento de consultas. Por isso, a questão de desempenho se tornou essencial para os usuários de BD devido à necessidade de respostas rápidas.

As técnicas de banco de dados implementadas nos BDD são as mesmas implementadas em BD centralizados, sendo que em BDD deve-se tratar a questão da distribuição dos dados durante o processamento de consultas. Em um ambiente de BD centralizados, a escolha da melhor estratégia de execução da consulta é feita selecionando a melhor consulta de álgebra relacional. No entanto, em um ambiente de BDD, além da melhor consulta de álgebra relacional, é preciso levar em conta o custo das operações para intercâmbio de dados entre os computadores e o melhor local para processar a consulta. A operação denominada semi-junção é uma técnica utilizada no processamento de consultas em BDD para a redução de tuplas de uma relação antes de enviá-la a outro computador, diminuindo assim o custo envolvido com a troca de dados.

Com o crescimento do uso de computadores pessoais (cada vez melhores e baratos) em um ambiente distribuído, surgiu o processamento paralelo com a finalidade de prover maior disponibilidade e alto desempenho no processamento de consultas. No lugar de cooperação entre os computadores para a realização de uma tarefa, as tarefas passaram a ser executadas em paralelo (ao mesmo tempo). O processamento paralelo é realizado através de sistemas distribuídos em vários computadores (com arquiteturas paralelas com processadores fortemente acoplados) e conectados entre si através de uma rede rápida, realizando tarefas simultaneamente. Esse conjunto de computadores é denominado Agrupamento de Computadores [36]. VALDURIEZ [43] propõe a extensão desse conceito para os BD distribuídos, originando então o BD paralelo. Banco de Dados Paralelo é um sistema que combina o gerenciamento de BD e o processamento paralelo para aumentar o desempenho e a disponibilidade [27].

Em BD paralelo, a fase de otimização do processamento de consultas tira proveito da localização dos dados e dos recursos de arquitetura paralela para executar várias consultas simultaneamente através de transações concorrentes. Na literatura existem duas estratégias para implementação da fase de otimização: a primeira estratégia, apresentada por HONG e STONEBRAKER [12] e KABRA e DEWITT [17], consiste em dividir essa fase em duas partes: otimização serial e paralelização. Na otimização serial o plano de execução é escolhido sem levar em conta o ambiente paralelo e na paralelização o plano de execução escolhido é modificado adicionando parâmetros do ambiente paralelo; a segunda estratégia, descrita por BARU *et al.* [2] consiste em manter o otimizador atualizado com informações sobre o custo de processamento e de paralelização, de modo que ele utilize as informações conhecidas no momento da escolha do melhor plano de execução da consulta.

Existem dois tipos de paralelismo para o processamento de consultas [27]: intra-consulta e inter-consulta. O paralelismo intra-consulta permite que uma mesma consulta seja executada por vários computadores; e o paralelismo inter-consulta permite que várias consultas sejam executadas ao mesmo tempo.

O uso de BD paralelos em agrupamentos de computadores para obter processamento paralelo, levou AKAL *et al.* [1] a criar o conceito de agrupamento de BD e a desenvolver uma nova estratégia para aumento de desempenho. Agrupamento de Banco de Dados é um conjunto de computadores (nós), que se comunicam através de uma rede, onde cada computador possui uma instância de um SGBD (de código aberto) e um aplicativo central para gerenciar esse conjunto de SGBD.

Embora OLAP seja uma tecnologia com a finalidade de gerar respostas rápidas, pois visa auxiliar a tomada de decisão, essas consultas tendem a serem custosas devido à sua natureza, à complexidade, ao número de agregações e filtros e ao grande volume de dados na base envolvida. Esse custo está relacionado ao tempo de processamento dessas consultas [26], que podem levar horas ou dias. Face ao exposto, obter um bom desempenho durante o processamento de consultas OLAP é um requisito físico essencial para o uso desta tecnologia. E o uso de paralelização no processamento de consultas OLAP é uma estratégia viável para o aumento de desempenho.

Aliando as técnicas de processamento paralelo de consultas ao agrupamento de BD, algumas soluções foram propostas para aumentar o desempenho de consultas OLAP através de paralelismo. Essas soluções serão detalhadas na seção 2.7 deste capítulo.

2.4 Fragmentação Virtual

O processamento paralelo intra-consulta permite que uma mesma consulta seja executada por vários computadores simultaneamente. ÖSZU e VALDURIEZ [27] descrevem dois tipos de paralelismo intra-consulta: entre-operadores ou intra-operador. No paralelismo entre-operadores vários operadores de uma consulta são executados por diferentes processadores, ou seja, cada operador de uma consulta é executado por um processador. No paralelismo intra-operador, o mesmo operador é executado por vários processadores, ou seja, um operador é decomposto em vários sub-operadores independentes e executados por diferentes processadores. Ambos os tipos de paralelismo são realizados em uma base de dados distribuída, e os vários operadores ou sub-operadores são executados sobre cada fragmento de relação em um computador.

Uma das abordagens utilizadas para se obter paralelismo intra-consulta é a Fragmentação Virtual – FV, que em vez de decompor o mesmo operador em vários sub-operadores, decompõe-se a consulta em várias sub-consultas para serem processadas em uma determinada porção de dados. A decomposição de uma consulta em sub-consultas é feita adicionando um predicado que define o intervalo de dados onde a sub-consulta será processada, ou seja, cada sub-consulta é processada por um computador em uma porção de dados diferente. Para possibilitar o processamento de uma sub-consulta por qualquer computador, evitando a verificação do local de dados necessários, o BD é totalmente replicado nos computadores. Essa abordagem foi proposta por AKAL *et al.* [1] para se obter paralelismo intra-consulta em um agrupamento de BD.

A partir de então, surgiu o conceito de Fragmento Virtual, que foi definido por LIMA [21] como sendo uma sub-relação $\mathbf{R}_{vp}$ resultante de uma seleção sobre uma relação $\mathbf{R}$ com um predicado $\mathbf{P}$ na forma de $\mathbf{a}_1 < \mathbf{A} \leq \mathbf{a}_2$, $\mathbf{a}_2 > \mathbf{a}_1$, onde $\mathbf{A}$ é um atributo de fragmentação de $\mathbf{R}$ e $\mathbf{a}_1$ e $\mathbf{a}_2$ são valores do domínio de $\mathbf{A}$.

A FV consiste em enviar a mesma consulta para todos os nós do agrupamento de BD, sendo que cada consulta será processada sobre um fragmento virtual diferente. Isso é feito substituindo a relação $\mathbf{R}$ de uma consulta por um fragmento $\mathbf{R}_{vp}$.

AKAL *et al.* [1] propõe a seguinte estratégia de funcionamento da FV: primeiro escolhe-se o Atributo de Fragmentação Virtual - AFV, preferencialmente aquele que possui valores numéricos contínuos; em seguida, define-se o número de nós do agrupamento de BD para o processamento de consultas; e por último, divide-se o intervalo do domínio do AFV pelo número de nós.

Podemos exemplificar o funcionamento da FV utilizando uma consulta sobre uma relação (CD00AMPESS) do Censo Demográfico 2000[4], que informa o número de pessoas do sexo masculino e feminino no Estado do Rio de Janeiro. Seja uma consulta C e um agrupamento de BD com quatro nós.

```
C: select codufcenso as uf,
         codsexo as sexo,
         sum(cd00ampess.pesopess) as frequencia,
         count(*) as contagem
    from cd00ampess
    where codufcenso = 33
    group by codufcenso, codsexo;
```

A FV irá decompor a consulta C em quatro sub-consultas (C_1, C_2, C_3 e C_4), pois este é o número de nós do agrupamento, e irá atribuir cada uma dessas sub-consultas a um nó do agrupamento. O AFV da relação CD00AMPESS é o atributo CONTROLE e possui valores numéricos contínuos dentro do intervalo [1, 5.304.711]. Cada nó irá processar um intervalo de 1.326.178 valores (número obtido pela divisão do limite superior do intervalo pelo número de nós). A seguir apresentamos as sub-consultas reescritas, com a adição de um predicado que define o fragmento virtual a ser processado.

```
C1: select codufcenso as uf,
         codsexo as sexo,
         sum(cd00ampess.pesopess) as frequencia,
         count(*) as contagem
    from cd00ampess
    where codufcenso = 33 and controle between 1 and 1326178
    group by codufcenso, codsexo;

C2: select codufcenso as uf,
         codsexo as sexo,
         sum(cd00ampess.pesopess) as frequencia,
         count(*) as contagem
    from cd00ampess
    where codufcenso = 33 and controle between 1326179 and 2652356
    group by codufcenso, codsexo;

C3: select codufcenso as uf,
         codsexo as sexo,
         sum(cd00ampess.pesopess) as frequencia,
         count(*) as contagem
    from cd00ampess
    where codufcenso = 33 and controle between 2652357 and 3978534
    group by codufcenso, codsexo;

C4: select codufcenso as uf,
         codsexo as sexo,
         sum(cd00ampess.pesopess) as frequencia,
         count(*) as contagem
    from cd00ampess
    where codufcenso = 33 and controle between 3978535 and 5304711
    group by codufcenso, codsexo;
```

No exemplo acima, cada nó receberá o mesmo tamanho de fragmento virtual. Mas se os valores do AFV não forem contínuos dentro o intervalo dado (distorção de dados),

[4] O Censo Demográfico 2000 é uma pesquisa do IBGE que possui informações sobre a população do Brasil.

essa estratégia de FV pode causar um desbalanceamento na distribuição dos intervalos a serem processados, isto é, um computador pode receber um intervalo maior do que o intervalo recebido por outro computador, aumentando a sua carga de processamento.

Existem alguns requisitos para o uso da FV no paralelismo de consultas. O primeiro deles é a replicação total do BD entre os computadores do agrupamento de BD, para garantir que qualquer computador possa processar qualquer sub-consulta sobre qualquer fragmento virtual. Segundo, é preciso que os valores do AFV de uma relação estejam agrupados e ordenados fisicamente, para aumentar a probabilidade de os dados que precisam ser lidos durante o processamento de uma sub-consulta estejam armazenados em um mesmo bloco de disco. O agrupamento e a ordenação física é obtida através de Índices de Agrupamento. Muitos SGBD implementam índices de agrupamento, que ao serem definidos sobre um atributo, ordenam a tabela fisicamente segundo esse atributo. E por último, é necessário ter um índice sobre o AFV, para evitar que o SGBD faça uma leitura em todos os registros da tabela (*full scan*), durante o processamento da consulta.

2.5 ParGRES

ParGRES [24, 29], originado do SmaQ [21], é uma camada intermediária entre um BD e uma aplicação cliente em um agrupamento de BD, cujo objetivo é aumentar o desempenho do processamento de consultas OLAP através de paralelismo inter-consulta e intra-consulta. A Figura 3 mostra a arquitetura geral de um agrupamento de BD utilizando o ParGRES. No esquema apresentado, uma aplicação cliente envia uma consulta ao ParGRES, que após analisá-la, decide se a mesma vai ser processada utilizando paralelismo inter- ou intra-consulta. Uma vez escolhido o tipo de paralelização, a consulta é enviada para o SGBD, que processa a consulta. Se a paralelização escolhida for a inter-consulta, o ParGRES envia a consulta para um dos nós processarem; se for intra-consulta, o ParGRES divide a consulta em várias sub-consultas e envia cada sub-consulta para um nó diferente.

Do recebimento ao processamento da consulta, quatro tarefas principais são executadas pelo ParGRES e serão descritas nas duas seções a seguir [21]:

1. tradução da consulta SQL;
2. processamento de consultas com paralelização (inter- ou intra-consulta);
3. composição de resultados;

4. processamento de atualização.

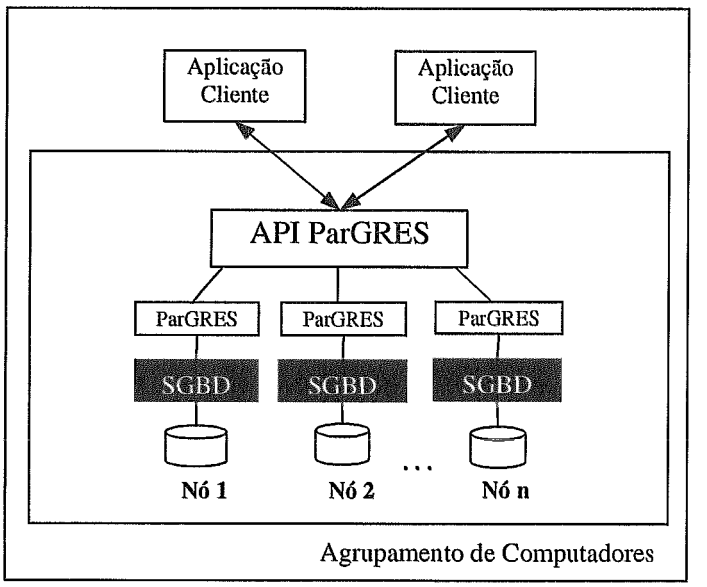


Figura 2: Arquitetura geral do ParGRES

2.5.1 Fragmentação Virtual Adaptativa

Antes de utilizar o ParGRES, é necessário configurar o número de nós disponível no agrupamento de BD para o processamento de consultas em paralelo. Ao receber o SQL de uma consulta, o módulo do sistema denominado Tradutor, realiza a identificação das relações e atributos que são utilizados nas agregações e os que podem ser utilizados no paralelismo intra-consulta. Essa tarefa de tradução da consulta SQL é realizada por um analisador sintático com uma gramática livre de contexto apropriada.

Após a análise da consulta e a geração das informações necessárias para o processamento das consultas, é preciso decidir o tipo de paralelismo a ser utilizado, dando início à tarefa de processamento de consultas com paralelização.

Se o ParGRES optar em processar a consulta utilizando paralelismo inter-consulta, ele verifica qual nó está menos sobrecarregado (aquele que possui o menor número de tarefas pendentes) e envia a consulta para ser processada. Se escolher o paralelismo intra-consulta, ele decompõe a consulta em várias sub-consultas e as envia para todos os nós (previamente configurados) do agrupamento de BD para serem processadas. A decomposição da consulta é feita utilizando a técnica de FV, descrita na seção 2.4, que consiste basicamente em dividir a consulta original em **n** sub-consultas, onde **n** equivale ao número de nós. Cada sub-consulta é reescrita, adicionando-se um

predicado com um determinado intervalo de valores (diferente para cada sub-consulta), e enviada para cada um dos nós, isto é, cada nó recebe uma dessas sub-consultas. Se o BD está totalmente replicado em todos os nós, o ParGRES pode enviar qualquer sub-consulta para qualquer nó processá-la.

Seja uma consulta Q sobre uma tabela qualquer:

```
select sum (price)
from part
where category = 'Y';
```

De forma a implementar a FV, a consulta será reescrita da seguinte maneira:

```
select sum (price)
from part
where category = 'Y'
and pid >= :v1 and pid < :v2
```

No exemplo acima, o atributo `pid` é escolhido como o AFV para a tabela `part`. Durante a reescrita da consulta, diferentes intervalos de valores (`v1` e `v2`) são atribuídos para cada nó do agrupamento de BD. Similarmente ao que é proposto em AKAL *et al.* [1], o ParGRES divide o valor máximo de `pid` pelo número de nós do agrupamento, e envia para cada um o mesmo número de linhas para serem processadas. Por exemplo, se o atributo `pid` da tabela possui o intervalo de valores `[1, 1000]` e quatro nós estão configurados no ParGRES para o processamento de consultas, o nó 1 receberá o intervalo `[1, 250]` para processar; o nó 2 receberá o intervalo `[251, 500]` para processar; e assim por diante, até alcançar o tamanho máximo do intervalo do AFV da tabela. O processamento de cada sub-consulta é feito localmente pelo SGBD de cada nó. Portanto, a FV é utilizada apenas na reescrita da consulta original em sub-consultas a serem enviadas para cada nó do agrupamento de BD.

Quando uma sub-consulta é enviada a um nó do agrupamento, ela não é executada imediatamente pelo SGBD do nó, mas dividida novamente em intervalos bem pequenos, a fim de aumentar o desempenho de processamento e prover maior flexibilidade durante o balanceamento de carga entre os nós.

A técnica utilizada para dividir uma sub-consulta localmente em um nó (em intervalos bem pequenos), é uma variação da FV e é denominada Fragmentação Virtual Adaptativa – AVP (*Adaptive Virtual Partitioning*), proposta por LIMA *et al.* [22]. A AVP consiste em dividir dinamicamente a sub-consulta, adaptando os intervalos a serem processados de acordo com o tempo de processamento dos intervalos já processados (naquele mesmo nó).

Então, ao receber uma sub-consulta, o nó realiza uma nova fragmentação dessa sub-consulta, através da divisão do intervalo do AFV (que ele recebeu) em intervalos bem menores. Em seguida, inicia o processamento desses novos intervalos, adaptando os seguintes (aumentando ou diminuindo) de acordo com o tempo de processamento dos intervalos, à medida que vão sendo processados. Esses intervalos se mantêm pequenos e adaptáveis a fim de se evitar uma leitura de todos os registros da tabela (*full scan*).

A AVP requer uma fase extra de processamento para compor o resultado final a partir dos resultados parciais gerados por cada nó. Para a realização da tarefa de composição de resultados, o ParGRES utiliza o HSQLDB [14], um SGBD de código livre e que utiliza poucos recursos de máquina, para criar uma tabela temporária onde os resultados parciais serão inseridos. Ao fim do processamento de todos os nós, o ParGRES realiza a composição do resultado final, recuperando os resultados parciais dessa tabela temporária. Depois de compor o resultado final e enviá-lo à aplicação cliente, a tabela temporária é eliminada.

2.5.2 Balanceamento de carga

A idéia de manter diversas sub-consultas sendo processadas por cada nó, aliada a replicação total do BD, permite o ParGRES realizar balanceamento de carga entre os nós de forma otimizada e dinâmica (durante o processamento da consulta). Esse balanceamento é feito através da redistribuição de carga, ou seja, atribuindo os fragmentos virtuais de um nó que ainda não foram processados para outro nó processá-los.

A abordagem do balanceamento de carga no ParGRES é feita através de ofertas de ajuda entre os nós, que se baseia na organização lógica dos nós e em um mecanismo de difusão de mensagens [21]. As mensagens de oferta de ajuda são aquelas enviadas por um nó (ofertante), quando se torna livre, oferecendo-se para o processamento de sub-consultas que ainda não foram processadas. As mensagens de aceite de ajuda são aquelas enviadas por um nó que aceita a ajuda e envia as sub-consultas que ainda não foram processadas.

Os nós são organizados logicamente em uma malha, seguindo o conceito de grafo-d dimensional em malha [44], onde o nó possui, no máximo, quatro vizinhos: norte, sul, leste e oeste. LIMA [21] define um grafo-d dimensional em malha como sendo um grafo que possui $n_0 * n_1 * \ldots * n_{d-1}$ vértices, identificados por um, e

somente um, elemento de $\{(i_0, i_j, \ldots, i_{d-1}) \mid 0 \leq i_j < n_j, 0 \leq j < d\}$, e arestas conectando os vértices, diferindo em um, e exatamente um, componente dos seus identificadores. O uso desse tipo de organização lógica considera que é permitida a comunicação direta entre os nós do agrupamento de computadores.

A Figura 4 ilustra um grafo-d dimensional em malha. Por exemplo, o nó "2,2" possui como vizinhos norte, sul, leste e oeste, os nós "1,2", "3,2", "2,3" e "2,1", respectivamente.

Uma vez definida a organização dos nós, a difusão de mensagens é feita de forma que um nó não receba uma mensagem mais de uma vez e o nó que originou a mensagem saiba quando todos os nós a receberam. O objetivo é evitar trabalho redundante e saber quando reiniciar a difusão de mensagens.

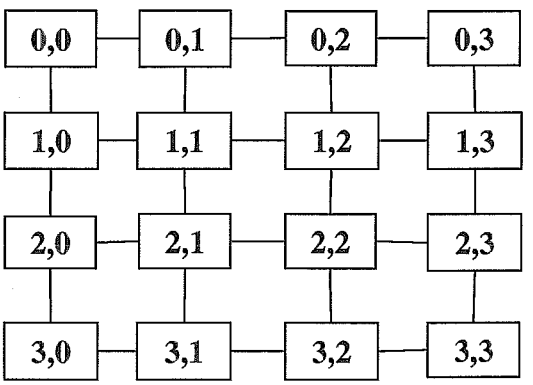


Figura 3: Grafo-d dimensional em malha. **Fonte**: modificado de LIMA[21].

O seguinte algoritmo [21] é utilizado por um nó para a difusão de mensagens e o mesmo é ilustrado na Figura 5:

1. se a mensagem é recebida de um nó vizinho do lado norte ou sul, repasse-a ao nó vizinho do lado oposto e para os nós vizinhos do lado leste e oeste.
2. se a mensagem é recebida de um nó vizinho do lado leste ou oeste, repasse-a ao nó vizinho do lado oposto.
3. se o nó se encontra na extremidade do lado leste (ou oeste) da malha, e recebe uma mensagem do vizinho do lado oeste (ou leste), avise ao nó que originou a mensagem de que não é possível mais propagar a mensagem.

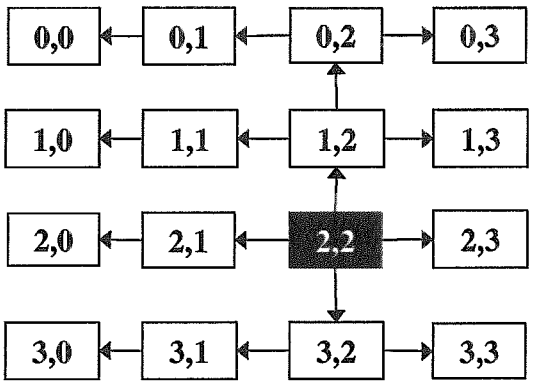


Figura 4: Ilustração do algoritmo de difusão de mensagens. **Fonte**: modificado de LIMA[21].

Quando um nó termina o processamento de todas as suas sub-consultas e se torna disponível, ele envia uma mensagem para os outros nós oferecendo ajuda para processar as suas sub-consultas, entrando em estado de espera por alguma resposta. Se um nó recebe uma mensagem de oferta de ajuda e está livre, ele repassa a mensagem para os outros nós. Se um nó recebe uma mensagem de oferta de ajuda e está ocupado, e não está processando o último intervalo, ele envia uma mensagem aceitando a ajuda para o nó ofertante. O nó ofertante interrompe seu estado de espera e solicita o intervalo a ser processado, que é enviado pelo nó que aceitou a ajuda. Se outras mensagens de aceite de ajuda chegar ao nó ofertante, essas serão descartadas.

A replicação total permite que um nó envie parte de suas sub-consultas não processadas para outro nó processar, porque todos possuem a mesma base de dados. Esse é um dos motivos o qual o balanceamento de carga do ParGRES é eficaz, pois há apenas tráfego de mensagens entre os nós, e não tráfego de dados.

2.5.3 Arquitetura

O ParGRES é formado por um conjunto de componentes, globais e locais, responsáveis pelas diversas etapas do processamento de consultas. Os componentes globais são o Processador de Consultas do Cluster – CQP (*Cluster Query Processor*), o Intermediador, o Tarefa de Consulta Global – GQT (*Global Query Task*) e o Coletor Global de Resultados – GRC (*Global Result Collector*). Os componentes locais são o Processador de Consultas de Nó – NQP (*Node Query Processor*), o Tarefa de Consulta Local – LQT (*Local Query Task*), o Executor de Consultas – QE (Query Executor) e o Coletor Local de Resultados – LRC (*Local Result Collector*).

Em um agrupamento de computadores não é permitida a conexão direta de um computador externo aos seus nós. Nesse tipo de ambiente existe um nó responsável (nó

de entrada) pela comunicação dos nós com o meio externo. Devido a essa característica do agrupamentos de computadores, o Intermediador é um componente do ParGRES, presente no nó de entrada, responsável pela comunicação das aplicações cliente e o ParGRES. Ao receber a consulta, o Intermediador a repassa para o CQP, também presente neste nó. Para cada consulta recebida, o CQP cria um novo GQT, no nó menos sobrecarregado, que é executado em uma linha de execução (*thread*) em separado. O GQT é responsável pela coordenação da execução da consulta, pela fase de Fragmentação Virtual Inicial e pela fase de encerramento do processamento da consulta. Para a realização da fragmentação virtual inicial, o GQT consulta os metadados do Catálogo para verificar os atributos de fragmentação e calcular os limites dos intervalos dos fragmentos iniciais. Após a fragmentação virtual inicial, o GQT cria um GRC, executado também em uma linha de execução (*thread*) em separado. O GRC é responsável pelo armazenamento e composição dos resultados parciais gerados por cada nó. Em seguida, o GQT envia para cada NQP uma sub-consulta a ser processada.

Em cada nó do agrupamento de BD há um NQP, responsável pelo processamento das sub-consultas, pela fase de ajustes do fragmento virtual e pela redistribuição de carga. Ao receber uma sub-consulta, o NPQ cria um LQT, que gerencia o processamento de uma consulta localmente. A consulta é enviada para o LQT, que cria o QE e o LRC, componentes responsáveis pela submissão da consulta ao SGBD e pela composição dos resultados em nível local, respectivamente. Esses componentes são criados em linhas de execução (*thread*) em separado. Antes de submeter a consulta, o QE cria um Modulador de Fragmentos, onde é implementado o algoritmo da AVP. Durante o processamento da consulta, o QE e o Modulador interagem entre si para trocarem informações sobre as consultas sendo processadas, resultados das consultas processadas e tempo gasto no processamento da consulta. É com base no tempo de processamento de um fragmento que o Modulador calcula o próximo intervalo do fragmento a ser processado. Os resultados de cada sub-consulta é armazenado pelo LRC, que ao final do processamento, compõem o resultado e envia para o GRC.

Após receber os resultados de todos os nós, o GRC compõe o resultado da consulta e o envia para o CQP, que o envia para o Intermediador e finalmente, o envia para a aplicação cliente (que originou a consulta). A Figura 6 ilustra a arquitetura dos componentes globais e locais.

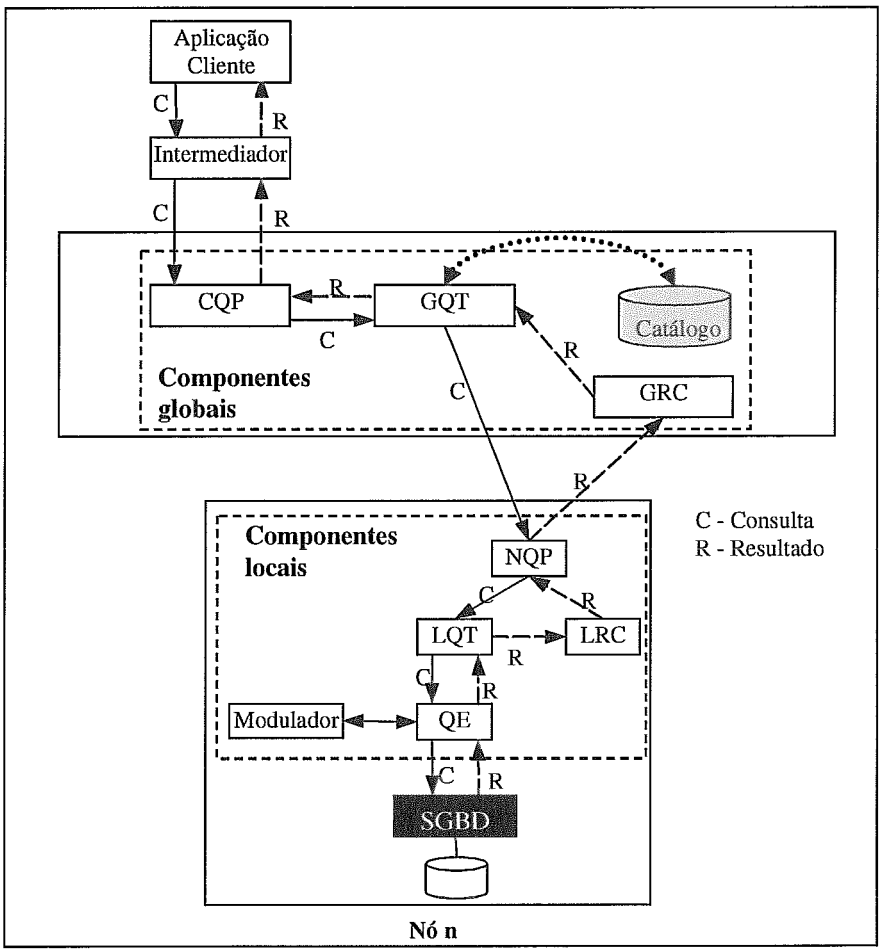


Figura 5: Arquitetura dos componentes globais e locais

2.6 SmaQSS

Apesar dos bons resultados obtidos pelo ParGRES no que tange a otimização do processamento de consultas, o uso de uma base totalmente replicada em todos os nós do agrupamento para obter paralelismo intra-consulta, pode tornar inviável a sua utilização. A fim de eliminar a necessidade da replicação total de uma base, FURTADO [10] propôs o SmaQSS (*Smashing Queries Shrinking Space*). O SmaQSS é um protótipo originado do SmaQ [21], que une a AVP e a replicação parcial de um BD para obter paralelismo intra-consulta no processamento de consultas OLAP.

2.6.1 Fragmentação Virtual Híbrida

A Fragmentação Virtual Híbrida – AHP (*Adaptive Hybrid Partitioning*) é resultado da combinação da AVP e a Fragmentação Híbrida – HP (*Hybrid Partitioning*).

A HP é uma abordagem que combina a fragmentação física das tabelas maiores e mais acessadas e a replicação total das pequenas tabelas [34].

A primeira etapa da AHP consiste na realização da HP sobre a base de dados, fragmentando as tabelas de fatos e replicando totalmente as tabelas de dimensão. As tabelas de fatos são fragmentadas através da operação de seleção sobre um atributo (mesma operação realizada pela Fragmentação Horizontal) e de acordo com o número de nós do agrupamento de BD. O atributo escolhido para a realização da fragmentação física é o mesmo atributo que será utilizado como AFV, e o predicado de seleção define um intervalo de valores do atributo de fragmentação. O número de fragmentos físicos gerados equivale ao número de nós e cada um dos fragmentos são alocados em um nó. Por exemplo, suponha um agrupamento de BD com quatro nós e uma relação `R` com mil linhas, cujo atributo escolhido para a fragmentação contenha valores numéricos sequenciais. Serão gerados quatro fragmentos dessa relação {Fg_1, Fg_2, Fg_3 e Fg_4}, cada um com 250 linhas e os intervalos de cada fragmento variando entre `[1..250]`, `[251..500]`, `[501..750]` e `[751..1000]` respectivamente. Assim, teremos a seguinte distribuição dos fragmentos entre os nós, ilustrada na Figura 7:

Nó	**1**	**2**	**3**	**4**
Fragmento	Fg_1[1..250]	Fg_2[251..500]	Fg_3[501..750]	Fg_4[751..1000]

Figura 6: Disposição dos fragmentos gerados pela HP entre os nós do agrupamento de BD

Uma das limitações do uso da HP como esquema de fragmentação física é a falta de replicação dos fragmentos gerados, o que dificulta o balanceamento de carga entre os nós. Por não ter um esquema de replicação, a HP gera apenas uma cópia do fragmento e a aloca em um nó. Para um nó ajudar o outro e assim realizar o balanceamento de carga, seria necessária a transferência de dados, o que torna uma solução inviável. Para contornar esse problema, FURTADO [10] propõe um modelo de replicação de dados com base no conceito de particionamento encadeado de HSIAO e DEWITT [13], descrito na seção 2.2.2. Nesse modelo, os fragmentos são replicados nos nós seguintes e adjacentes ao nó onde está armazenada a cópia primária. Se existe uma réplica para cada cópia primária, essa réplica é armazenada no nó seguinte e adjacente; se existem duas réplicas, essas são armazenadas nos dois nós seguintes e adjacentes; e assim sucessivamente. As Figuras 8 e 9 ilustram esse modelo de replicação.

A arquitetura do SmaQSS é semelhante à do ParGRES (apresentada na seção 2.5.3), e possui os mesmos componentes locais e globais do processamento de consulta.

No entanto, algumas modificações foram necessárias, no que tange o funcionamento de alguns componentes, a fim de oferecer suporte ao uso de base parcialmente replicada. No ParGRES, as informações existentes no Catálogo são a respeito das tabelas e atributos envolvidos na AVP. No SmaQSS, além dessas informações, são necessárias informações sobre os intervalos de cada cópia primária das relações e a localização das mesmas, bem como informações sobre as réplicas. Quando o Intermediador recebe a consulta da aplicação cliente, essa é repassada para o CQP, que inicia a fase da fragmentação virtual inicial, tal como é feito no ParGRES. A etapa de envio das sub-consultas para os nós é diferente no SmaQSS. Enquanto no ParGRES não há a preocupação sobre quais sub-consultas serão atribuídas para cada nó, no SmaQSS, as sub-consultas são enviadas para o nó que pode processá-las, ou seja, antes de enviar uma sub-consulta, é verificado qual nó possui o fragmento físico do qual o intervalo da sub-consulta está incluído. Nem todo nó pode processar uma determinada sub-consulta.

Nó	1	2	3	4
Cópia Primária	Cp_1	Cp_2	Cp_3	Cp_4
Réplica 1	Rp_4	Rp_1	Rp_2	Rp_3

Figura 7: Modelo de Replicação para uma réplica

Nó	1	2	3	4
Cópia Primária	Cp_1	Cp_2	Cp_3	Cp_4
Réplica 1	Rp_4	Rp_1	Rp_2	Rp_3
Réplica 2	Rp_3	Rp_4	Rp_1	Rp_2

Figura 8: Modelo de Replicação para duas réplicas

2.6.2 Balanceamento de carga e Políticas de ajuda

A abordagem do balanceamento de carga do SmaQSS é a mesma utilizada pelo ParGRES, baseada em difusão de mensagens, mas ao contrário de como o balanceamento é tratado no ParGRES, o SmaQSS utiliza base parcialmente replicada e a organização lógica dos nós segue o mesmo modelo da replicação por particionamento encadeado. O nó ofertante tem um prévio conhecimento dos nós que ele pode ajudar, então a mensagem de oferta de ajuda será enviada apenas para os nós que possuem os seus mesmos intervalos de dados. Face ao exposto, na Figura 8, o nó 2 pode ajudar o nó 1 e o nó 3; e na Figura 9, o nó 2 pode ajudar todos os nós. Essa nova abordagem reduz o número de mensagens que trafegam pela rede do agrupamento de BD.

A definição de políticas de ajuda foi outra alteração realizada no algoritmo de balanceamento de carga. No ParGRES, não há nenhum tratamento no recebimento de mensagens de aceite de ajuda. Se o nó ofertante já está processando uma nova sub-consulta de outro nó, qualquer mensagem recebida é descartada. No SmaQSS foram definidos quatro tipo de políticas de ajuda, que têm por objetivo manter uma fila ordenada de todas as mensagens de aceite que o nó ofertante recebe durante o processamento de qualquer sub-consulta. Essas políticas de ajuda são baseadas na ordem de chegada das mensagens de aceite de ajuda ou na carga que ainda falta ser processada pelo nó que aceitou a ajuda. As políticas são FIFO, LIFO, MWLF e LWLF [10].

As políticas FIFO e LIFO levam em conta o momento em que a mensagem chega ao nó ofertante. A política FIFO (*First In First Out*), que traduzido significa "primeiro a chegar, primeiro a sair", organiza as mensagens de acordo com a ordem (crescente) de chegada da mensagem, ou seja, o nó que enviou a primeira mensagem de aceite recebida será o primeiro a ser ajudado. A política LIFO (*Last In First Out*), que traduzido significa "último a chegar, primeiro a sair" organiza as mensagens de acordo com a ordem (decrescente) de chegada da mensagem, ou seja, o nó que enviou a última mensagem recebida será o primeiro a ser ajudado.

As políticas MWLF e LWLF levam em conta a carga a ser processada pelo nó que aceitou a ajuda. A política MWLF (*Most Workloaded First*), que traduzido significa "mais sobrecarregado primeiro", organiza as mensagens de acordo com a sobrecarga do nó, ou seja, o nó mais sobrecarregado será o primeiro a ser ajudado. A política LWLF (*Least Workloaded First*), que traduzido significa "menos sobrecarregado primeiro", organiza as mensagens de acordo com a baixa carga do nó, ou seja, o nó menos sobrecarregado será o primeiro a ser ajudado.

Ao retirar a primeira mensagem da fila, o algoritmo de balanceamento de carga verifica se o nó que enviou a mensagem de aceite ainda está precisando de ajuda, solicita o intervalo e o processa. É possível que algum nó que enviou uma mensagem de aceite não precise mais de ajuda, pois a mensagem já estava há algum tempo na fila e o mesmo processou todo o seu intervalo (quando o nó envia uma mensagem de aceite de ajuda, ele continua processando suas sub-consultas até o momento em que o nó ofertante solicita o intervalo a ser processado). À medida que o nó ofertante vai ajudando os nós, ele retira as mensagens da lista, até esvaziá-la. Quando a fila fica vazia, ele envia nova mensagem de oferta de ajuda aos seus nós vizinhos.

Para reduzir o tempo desperdiçado de uma mensagem que fica na fila e o nó não precise mais de ajuda, foi criado um parâmetro configurável de forma que limite o tempo de uma mensagem em espera na fila. Assim, de tempos em tempos, é verificado o tempo de espera de cada mensagem na fila, e caso esse tempo tenha sido ultrapassado, a mensagem é retirada da fila. O algoritmo de balanceamento de carga é executado até que todos os fragmentos virtuais tenham sido processados.

2.7 Trabalhos correlatos

Existem na literatura diversas soluções para a otimização de desempenho no processamento paralelo de consultas em agrupamento de banco de dados, como o PowerDB [31], C-JDBC [6], Apuama [25] e Sequoia [35]. Vale salientar que apenas o ParGRES e o PowerDB oferecem paralelismo intra-consulta, no entanto, o PowerDB apresenta diversas limitações no que tange o desbalanceamento de carga entre os nós durante o processamento de consultas. A seguir descreveremos cada uma dessas soluções.

2.7.1 PowerDB

O PowerDB é uma camada intermediária entre o BD e a aplicação cliente em agrupamentos de BD, que através da replicação total de um BD, obtém processamento paralelo de consultas. Diversas soluções têm sido propostas dentro do PowerDB para prover paralelismo no processamento de consultas OLTP e OLAP. No que tange o processamento de consultas OLAP, AKAL *et al.* [1] propôs o uso de fragmentação virtual para obter paralelismo intra-consulta. Entretanto, a FV proposta apresenta limitações pois esta se baseia em uma distribuição de dados uniforme no AFV. A abordagem "uma sub-consulta por nó" é também uma questão que evita o balanceamento dinâmico de carga no PowerDB, pois os SGBD são usados como componentes do tipo "caixa-preta".

RÖHM *et al.* [34] propôs a Fragmentação Híbrida – FH, uma abordagem de projeto físico de BD que evita a replicação total da base, através da fragmentação das tabelas de fatos e da alocação de cada fragmento em um nó diferente do agrupamento de BD. As tabelas de dimensões são totalmente replicadas entre os nós. A FH torna possível implementar o paralelismo intra-consulta durante o processamento de consultas OLAP. Porém, se o desbalanceamento de carga ocorrer durante a execução da consulta,

a transferência de dados deve ser feita a fim de redistribuir a carga. Nenhuma política de balanceamento de carga é proposto pelos autores.

Outros trabalhos foram feitos no contexto do PowerDB [32, 33] mas eles não empregam o paralelismo intra-consulta, o que não causa a redução do tempo de processamento individual de consultas, sendo imperativo para agilizar o processo de tomadas de decisão. Por fim, o PowerDB não é um programa de código-fonte aberto nem está disponível para ser copiado em seu sítio da Internet.

2.7.2 C-JDBC

O C-JDBC é uma camada intermediária entre o BD e a aplicação cliente em agrupamentos de BD desenvolvido por CECCHET *et al.* [6], que utiliza uma abordagem parecida com a do ParGRES. Ele atende, principalmente, as transações OLTP em aplicações de comércio eletrôncio, e utiliza replicação parcial e total de BD para prover paralelismo inter-consulta. Em [6] os resultados apresentados mostraram que o C-JDBC apresentou bom desempenho no processamento de consultas do *Benchmark* TPC-W [42]. Particularmente, a replicação parcial foi melhor do que a replicação total por causa das operações de atualização, muito comuns em aplicações de comércio eletrônico. Entretanto, por não oferecer suporte ao paralelismo intra-consulta, o C-JDBC não é adequado para o uso de processamento de consultas OLAP, pois o mesmo demonstrou satisfazer melhor às aplicações que resultam em consultas rápidas e concorrentes, como as existentes no TPC-C [40] e TPC-W [42]. Ao contrário do PowerDB, o C-JDB é um programa de código-fonte aberto.

2.7.3 Apuama

O Apuama é uma solução proposta por Miranda *et al.* [25] para adicionar paralelismo intra-consulta ao aplicativo C-JDBC. O paralelismo intra-consulta é implementado através de FV estática e replicação total de BD. Os resultados obtidos demonstraram que essa solução apresentou bom desempenho no processamento de consultas. Entretanto, essa solução não se comporta adequadamente em BD que são utilizados como componentes do tipo "caixa-preta", porque a mesma possui comandos específicos de SGBD para forçar o uso de índices de agrupamento durante o processamento da consulta e obter um bom desempenho com FV. Por ser muito sensível

a distorção de dados, não é possível realizar balanceamento de carga entre os nós utilizando o Apuama.

2.7.4 Sequoia

O Sequoia [35, 5], extensão do projeto C-JDBC, é uma camada intermediária entre o BD e a aplicação cliente, que permite qualquer aplicação Java acesse um agrupamento de BD através de JDBC. Ele implementa o RAIDb (*Redundant Array of Inexpensive Databases*), uma combinação de diversos BD de baixo custo, para dar suporte à paralelização, balanceamento de carga e tolerância a falhas. O RAIDb surgiu a partir do conceito de RAID (*Redundant Array of Inexpensive Disks*), uma tecnologia que obtém escalabilidade e disponibilidade, com baixo custo, através de um subsistema de discos. Utilizando replicação e distribuição de BD entre os nós, o Sequoia aumenta o desempenho do processamento de consultas e realiza o balanceamento de carga. No entanto, os resultados apresentados mostram que esse aplicativo alcança melhor desempenho em transações OLTP de aplicações de comércio eletrônico.

3 Paralelização de consultas OLAP com o ParGRES

Neste capítulo é apresentada uma sistematização de passos necessários para transformar uma aplicação baseada em consultas sequenciais OLAP em uma aplicação que possibilite consultas paralelas sobre bases de dados OLAP utilizando o ParGRES. Na seção 3.1 apresentamos uma metodologia para migração de uma base de dados centralizada para uma base de dados distribuída, da qual será utilizada com o ParGRES para a obtenção de paralelismo e alto desempenho no processamento de consultas; e na seção 3.2 descrevemos as alterações realizadas na versão atual do ParGRES para o seu funcionamento com bases parcialmente replicadas.

3.1 Metodologia para utilização do ParGRES

A utilização do ParGRES para a obtenção de paralelismo no processamento de consultas não acarreta qualquer alteração no projeto físico de uma base de dados durante a migração de um ambiente centralizado para um distribuído. No entanto, podemos tirar proveito do projeto físico existente e aumentar ainda mais o desempenho do processamento de consultas. Para a realização deste processo de migração de BD, desenvolvemos uma metodologia a partir de um conjunto de recomendações a serem seguidas para um melhor aproveitamento do uso do ParGRES na paralelização de consultas.

Essa metodologia consiste na realização de duas etapas principais, descritas nas próximas seções. Na seção 3.1.1 descrevemos a etapa de análise do esquema conceitual e do projeto físico do banco de dados a ser migrado; e na seção 3.1.2 descrevemos a etapa de fragmentação e distribuição de BD para o ParGRES.

3.1.1 Análise do esquema conceitual e projeto físico de Banco de Dados

Para o desenvolvimento dessa metodologia assumimos que a base de dados a ser migrada já possui um esquema e um projeto físico definido. Todo e qualquer banco de dados possui um esquema e um projeto físico. Um esquema é uma definição formal da estrutura do banco e dos dados nele contidos e podem ser definidos em três níveis: interno (ou esquema interno), conceitual (ou esquema conceitual) e externo (ou visão de usuário) [9]. No esquema conceitual são definidas as entidades, os tipos de dados, as conexões, as operações de usuários e as restrições, e é esse esquema que nos interessa

no processo de migração. Após a definição do esquema conceitual, o projetista de banco de dados deve empreender o projeto físico. O projeto físico é uma descrição formal da estrutura de armazenamento físico dos dados em um BD, que tem como base as definições presentes no esquema conceitual e o principal objetivo é desenvolver uma estrutura de dados apropriada ao armazenamento, garantindo um bom desempenho durante as operações realizadas (inserção, atualização e exclusão).

Parte dos SGBD's relacionais implementa as relações (ou entidades) do esquema conceitual como arquivos físicos (ou tabelas) no BD; os atributos das relações são implementados como atributos das tabelas; os atributos-chave passam a ser as chaves primárias; e os relacionamentos existentes entre as entidades do esquema são representados por chaves estrangeiras. Para a implementação de relações de um esquema multidimensional para um modelo de armazenamento, estão envolvidos dois tipos de tabelas: de fatos ou de dimensões. No que tange a implementação de atributos, atributos-chave e relacionamentos, os mesmos são feitos da forma descrita inicialmente.

Uma questão importante no projeto físico de BD são as decisões acerca da indexação das tabelas, que se refere aos índices (estruturas de acesso) utilizados para aumentar a velocidade do acesso aos dados. Os índices podem ser criados sobre qualquer atributo da tabela, entretanto, o desempenho das consultas depende de quais índices existem. Por exemplo, em operações de inserção, atualização e exclusão, em vez de aumentar o desempenho, os índices aumentarão a sobrecarga dessas operações, pois a operação deverá ser realizada não só no atributo, mas também sobre o índice. Para criar um índice é recomendado que se conheça as consultas, as transações e as aplicações a serem executadas sobre o BD. Os atributos utilizados em predicados que possuem condições de igualdade ou de intervalos e os atributos que são chaves primárias ou compõem uma chave de junção são candidatas à criação de índices.

Os índices mais utilizados em projetos físicos de BD são os *B-tree*, *hash*, *bitmap* e de agrupamento (*clustered*). Os índices do tipo *B-tree* utilizam indexação de dados multinível e são um tipo especial de estrutura de dados de árvore. Uma árvore é formada por nós; e os nós (com exceção do nó raiz) são formados por um pai e vários filhos. Esses índices podem ser usados pelo otimizador de consultas do SGBD durante a realização de uma junção em uma consulta. Os índices do tipo *hash* utilizam uma função para localizar um registro; a função é aplicada sobre os valores do atributo indexado. O uso desses índices são recomendados quando os atributos utilizados nas consultas possuem condição de igualdade. Os índices do tipo *bitmap* utilizam um tipo

de indexação que constrói um vetor de bits para cada valor do atributo [**Erro! Fonte de referência não encontrada.**]. Recomenda-se a utilização desse tipo de índice para domínios de baixa cardinalidade, como é o caso das variáveis observadas em uma tabela de fato. Os índices de agrupamento (*clustered*) ordenam uma tabela fisicamente segundo um atributo, definido de acordo com a necessidade de se ordenar uma tabela por esse atributo. As consultas que possuem predicados com condições de intervalo, tiram vantagem deste índice, porque uma vez que o índice identifica o bloco de disco onde está armazenado o primeiro valor do intervalo, todas as outras linhas provavelmente estarão no mesmo bloco, então o acesso a disco e o tempo de execução da consulta serão minimizados. Em uma tabela só é permitida a criação de um único índice de agrupamento.

Alguns requisitos básicos de otimização em banco de dados OLAP devem ser atendidos para que o SGBD possa tirar proveito e obter o melhor desempenho durante a realização de consultas. Por esse motivo, a etapa de análise do esquema conceitual e do projeto físico do banco de dados torna-se essencial no processo de migração de base centralizada para distribuída, pois é recomendado que se mantenha o projeto existente para garantir um desempenho prévio do BD durante o uso do ParGRES.

Neste sentido, devem ser mantidos as seguintes características do projeto existente:

a) organização física de dados;
b) chaves primárias e chaves de junção;
c) chaves estrangeiras; e
d) índices.

Existem apenas dois requisitos obrigatórios para a utilização do ParGRES:

i) o uso de um atributo de fragmentação virtual (AFV); e
ii) a criação de um índice de agrupamento.

Conforme descrito na seção 2.4, o Atributo de Fragmentação Virtual – AFV, é um atributo da tabela que possui valores numéricos contínuos que será utilizada pela fragmentação virtual. Em geral, escolhe-se como AFV o atributo que compõe a chave primária da tabela, porque na maioria das vezes ele possui valores sequenciais contínuos. Se os atributos da tabela candidatos à AFV não possuem valores sequenciais contínuos, essa fragmentação pode causar uma má distribuição física dos dados, e

consequentemente, um desbalanceamento de carga durante a fragmentação virtual feita pelo ParGRES. No entanto, foi demonstrado em LIMA [21] o bom desempenho do ParGRES em uma base que apresenta distorção de dados, o que possibilita o uso de qualquer atributo como AFV.

Para tirar proveito dos valores sequenciais contínuos do AFV e prover uma correta fragmentação virtual, deve ser criado um índice de agrupamento sobre o atributo escolhido como AFV. É importante salientar que a falta de um índice de agrupamento pode acarretar a degradação do desempenho do processamento de consultas.

3.1.2 Projeto de fragmentação e distribuição do Banco de Dados no ParGRES

Para utilizar o ParGRES em uma base totalmente replicada não é necessário um projeto de fragmentação, pois a base de dados deve ser replicada em todos os nós do agrupamento de computadores. A utilização do ParGRES em uma base parcialmente replicada requer a definição de um projeto de fragmentação e distribuição dos dados.

Ambos os projetos, de fragmentação e distribuição, basearam-se nas estratégias de fragmentação e replicação propostas por FURTADO [10].

O projeto de fragmentação de um banco de dados no ParGRES consiste em particionar fisicamente uma tabela em relação ao número de nós. Ou seja, divide-se o número de registros de uma tabela (que está sendo fragmentada) pelo número de nós a ser utilizado para o processamento de consultas. O número de fragmentos é sempre igual ao número de nós e a cardinalidade de um fragmento em uma determinada configuração de nós é:

$$|F_n| = NTR / n$$

tal que:

NTR = Número Total de Registros

n = número de nós

A fragmentação física é feita de forma semelhante à fragmentação virtual, isto é, utilizando um atributo de fragmentação como predicado de seleção de tuplas. Recomenda-se o uso do atributo de fragmentação virtual escolhido na etapa anterior como atributo da fragmentação física. Para a definição dos fragmentos de uma tabela é interessante que este atributo contenha valores sequenciais uniformes (para garantir que o valor máximo desse atributo seja igual ao número de registros), para evitar um

desbalanceamento na distribuição física de dados (fragmentos com diferença significativa no número de registros). Neste sentido, os fragmentos são representados da seguinte forma:

$$F_n = ((n-1)((max(AFV))/n)+1) \leq AFV \leq (n((max(AFV))/n))$$

tal que:

AFV = Atributo de Fragmentação Virtual

n = número de nós

É importante lembrar que nem sempre o atributo escolhido como AFV terá valores sequenciais contínuos. Se os atributos da tabela candidatos à AFV não possuem valores sequenciais contínuos, essa fragmentação pode causar uma má distribuição física dos dados, acarretando um desbalanceamento de carga durante o processamento, que não poderá ser tratado pelo ParGRES. Neste caso, é recomendado a criação de um novo atributo na tabela, com a finalidade única de atuar como AFV, e que valores sequenciais sejam inseridos nesse atributo.

No modelo lógico (estrela) de um armazém de dados, as relações são representadas pelas tabelas de fatos e de dimensões. As tabelas de fatos desse modelo localizam-se no centro da estrela e essas tabelas armazenam o maior número de registros. Em geral, as tabelas de dimensão são bem pequenas, em número de linhas, quando comparadas com as tabelas de fato. Por esse motivo, a fragmentação física de uma base de dados OLAP é realizada apenas sobre as tabelas de fatos, pois elas possuem alta cardinalidade.

A estratégia de replicação de fragmentos consiste em distribuir os fragmentos por espalhamento encadeado. Essa distribuição é feita alocando uma réplica $\mathbf{Rp_n}$ de uma cópia primária (fragmento) $\mathbf{Cp_n}$ no nó seguinte $(\mathbf{n+1})$ à localização da cópia primária.

3.2 Arquitetura do ParGRES com Base Parcialmente Replicada

Conforme apresentado na seção 2.5, o ParGRES obtém paralelismo intra-consulta apenas em bases totalmente replicadas. Para avaliar o seu comportamento em bases parcialmente replicadas, adaptamos o seu funcionamento a esse novo contexto através de modificações em seu código-fonte, emulando parte do funcionamento do SmaQSS [10], descritas a seguir:

Definição da localização dos fragmentos. Durante o processo de fragmentação virtual, uma consulta é dividida em sub-consultas e enviadas para os nós processá-las (uma sub-consulta por nó). Em uma base totalmente replicada qualquer nó pode processar qualquer sub-consulta, pois todos os nós possuem todos os dados da base; porém, em uma base parcialmente replicada isso não acontece, pois cada nó tem apenas uma porção dos dados. Para definir quais nós podem processar um determinada sub-consulta, gerada durante o processo de fragmentação virtual, é necessário o prévio conhecimento da localização dos fragmentos (cópias primárias e réplicas) e os intervalos de dados contidos em cada um.

No ParGRES, implementamos a localização dos fragmentos, e seus respectivos intervalos de dados, utilizando a forma de geração desses fragmentos descrita na seção anterior. Por estar fragmentada, não é possível para o ParGRES conhecer o número total de registros nem o valor máximo do AFV, logo essa informação deve ser passada pelo usuário. Ao calcular os fragmentos existentes, cada fragmento recebe um número que o identifica, igual ao número de identificação do nó. Se forem gerados quatro fragmentos com 100 registros cada, assumimos que existem quatro nós no agrupamento, e que o Fragmento 1 está localizado no nó 1 e possui o intervalo de dados [1-100]; o Fragmento 2 está localizado no nó 2 e possui o intervalo de dados [101-200]; e assim sucessivamente. E as réplicas estão localizadas nos nós seguindo o espalhamento encadeado, também descrito na seção anterior.

Nova organização dos nós. Em uma base totalmente replicada qualquer nó pode ajudar outro nó a processar uma consulta, por isso a oferta de ajuda deve ser enviada a todos os nós do agrupamento. A difusão de ajuda é feita através de um mecanismo de propagação de mensagens que se baseia na organização lógica dos nós, seguindo o conceito de grafo-d dimensional em malha (descrito na seção 2.5.2). Em uma base parcialmente replicada, apenas alguns nós podem ajudar outros nós, de acordo com as réplicas que ele armazena. Portanto, um nó deve enviar oferta de ajuda apenas para aqueles nós os quais ele pode ajudar. Para garantir que apenas os nós que podem ser ajudados pelo nó ofertante receba a oferta de ajuda, implementamos uma nova organização lógica dos nós no ParGRES, seguindo o conceito de listas circulares. Nesta nova organização, o nó possui 2 vizinhos: leste e oeste.

Difusão de mensagens. De forma diferente à difusão de mensagens existente (descrita na seção 2.5.2), implementamos no ParGRES o envio de mensagens unidirecional baseado no espalhamento encadeado do projeto de replicação descrito na

seção anterior. Ou seja, um nó envia oferta de ajuda para os nós anteriores (oeste) ou para os nós seguintes (leste), de acordo com a direção em que as réplicas foram armazenadas nos nós da lista circular, de forma que a oferta de ajuda seja enviada apenas para os nós que possuem as cópias primárias das suas réplicas. Portanto, o número de nós que podem ajudar um determinado nó é sempre igual ao número de réplicas. Em nossa implementação assumimos que as réplicas foram armazenadas sempre nos vizinhos ao leste. Se o número de réplicas é igual a dois e essas foram armazenadas nos nós seguintes ao nó que armazena a cópia primária, o nó ofertante pode oferecer ajuda para os dois nós anteriores (nós que armazenam as cópias primárias das réplicas do nó ofertante).

Balanceamento de carga. Quando um nó acaba o processamento de uma consulta e se torna livre, ele envia oferta de ajuda aos nós que ainda estão processando. Quando um nó aceita ajuda, ele envia para o nó ofertante uma consulta com um determinado intervalo. Ao receber a consulta, o nó ofertante verifica se ele possui o intervalo de dados para processar a consulta recebida. Se sim, o nó ofertante assume a responsabilidade do intervalo recebido e processa a consulta. Se não, o nó ofertante envia uma mensagem para o nó que aceitou a ajuda informando que não vai processar a consulta.

Vale ressaltar que as alterações realizadas contemplam qualquer número de réplicas.

4 Projeto de fragmentação da base do Censo Demográfico 2000 e BME

O Instituto Brasileiro de Geografia e Estatística – IBGE [16] é o principal provedor de dados e informações sobre o Brasil, que atendem às necessidades dos mais diversos segmentos da sociedade civil, bem como dos órgãos das esferas governamentais federal, estadual e municipal. As principais funções desempenhadas pelo IBGE são:

- produção e análise de informações estatísticas;
- coordenação e consolidação das informações estatísticas;
- produção e análise de informações geográficas;
- coordenação e consolidação das informações geográficas;
- estruturação e implantação de um sistema da informações ambientais;
- documentação e disseminação de informações;
- coordenação dos sistemas estatístico e cartográfico nacionais.

Os dados e as informações providos pelo IBGE são obtidos através de levantamentos estatísticos de âmbito social, demográfico, econômico e geográfico. Dentre esses diversos levantamentos, destaca-se o Censo Demográfico, que se constitui como núcleo das estatísticas sociodemográficas. Por produzirem um grande volume de informações essenciais para a definição de políticas públicas e para a tomada de decisões de diversos segmentos (seja público ou privado), essa é uma das pesquisas muito demandadas por pesquisadores e usuários afins.

O IBGE possui um dos maiores acervos especializados em informações estatísticas e geográficas do país, que se constitui de publicações impressas e eletrônicas, como também de bases de dados.

A Internet é o principal canal de comunicação entre o IBGE e o usuário, onde estão disponíveis os resultados das pesquisas, seja na forma de páginas dinâmicas, arquivos para *download* ou bancos de dados. Nesse portal, encontra-se disponível o Banco Multidimensional de Estatísticas – BME, um armazém de dados desenvolvido pela Instituição, com a finalidade de prover um repositório de dados de suas pesquisas e

ferramentas de acesso a esses dados. O Censo Demográfico é uma das pesquisas presentes nesse sistema.

De acordo com o contexto descrito, temos um cenário típico de Sistemas de Informações Multidimensionais de Apoio a Decisão, com grande volume de dados e consultas de análise complexas e de alto custo. Assim, o Censo Demográfico 2000 e o Banco Multidimensional de Estatísticas são os elementos essenciais, junto com a tecnologia de agrupamento de BD (*cluster*) e processamento paralelo, para a criação de um ambiente real e ideal para a avaliação do ParGRES no processamento de consultas OLAP utilizando paralelismo intra-consulta. Portanto, a base de dados e as consultas OLAP utilizadas nos experimentos dessa dissertação são reais e oriundas do Censo Demográfico 2000 e do BME, respectivamente.

Neste capítulo são apresentados ao leitor o Censo Demográfico 2000, pesquisa realizada pelo Instituto Brasileiro de Geografia e Estatística – IBGE, e o Banco Multidimensional de Estatísticas - BME, um banco de dados disponível no portal do IBGE na Internet. Na seção 4.1 descrevemos o Censo Demográfico 2000, a granularidade e o volume de dados, quais e como as informações dessa pesquisa são divulgadas à sociedade; na seção 4.2 descrevemos o Banco Multidimensional de Estatísticas, o seu funcionamento, o modelo lógico e físico do Censo Demográfico no BME, a demanda de uso e custo da aplicação, e a escolha das consultas OLAP realizadas no BME para o experimento desta dissertação; na seção 4.3 apresentamos o projeto de fragmentação do Censo Demográfico 2000 no BME, as tabelas fragmentadas fisicamente, as chaves de fragmentação e demais informações relevantes do projeto de fragmentação; e por fim, na seção 4.4 apresentamos o projeto de fragmentação do Censo Demográfico 2000 no ParGRES, como foi feito o mapeamento da fragmentação dessa base no BME para uso no ParGRES, e as alterações necessárias para a realização dos experimentos.

Para simplificar a leitura desta dissertação, utilizaremos os termos CD2000 e BME para denominar o Censo Demográfico 2000 e Banco Multidimensional de Estatísticas, respectivamente.

4.1 Censo Demográfico 2000

O Censo Demográfico 2000 [4], pesquisa realizada pelo IBGE, é uma das principais fontes de informações sobre a situação de vida da população em cada um dos

municípios e localidades do país, produzindo informações fundamentais para a formulação de políticas públicas e a tomada de decisões de investimentos privados ou governamentais.

As informações coletadas e divulgadas são organizadas segundo a abrangência dos dados:

- Universo: informações sobre as características básicas do domicílio e dos seus moradores.
- Amostra: informações mais detalhadas sobre características do domicílio e de seus moradores, referentes aos temas religião, cor ou raça, deficiência, migração, escolaridade, fecundidade, nupcialidade, trabalho e rendimento, além das informações básicas (do Universo).

Os dados do Universo e da Amostra são organizados em temas: Domicílios, Pessoas e Famílias. Domicílio é o local estruturalmente separado e independente que se destina a servir de habitação a uma ou mais pessoas. No domicílio é considerada Família a pessoa que mora sozinha; o conjunto de pessoas ligadas por laços de parentesco ou de dependência doméstica; ou as pessoas ligadas por normas de convivência.

Durante o levantamento do Censo Demográfico 2000 foram investigados 54.265.618 domicílios, situados nos 5.507 municípios existentes naquele ano. Nesses domicílios as informações coletadas abrangeram 169.779.170 pessoas.

Os Censos Demográficos do Brasil utilizam a técnica de amostragem na coleta, que permite selecionar uma determinada porção do total de domicílios a serem investigados para responder ao questionário da amostra. As informações detalhadas desse questionário representam todo o universo de domicílios e de pessoas a partir da expansão dos dados coletados. Para realizar a expansão desses dados, são calculados pesos para cada um dos domicílios pesquisados, e esses pesos são atribuídos ao próprio domicílio e a cada um de seus moradores.

No Censo Demográfico 2000 foram selecionados 5.304.711 domicílios para responder ao questionário da amostra. As informações obtidas a partir desse questionário abrangeram 20.274.412 pessoas e 5.691.294 famílias.

4.1.1 Data de referência

A coleta dos Censos Demográficos ocorre decenalmente, ou seja, há um intervalo intercensitário máximo de dez anos. No Censo Demográfico 2000, a coleta de dados foi realizada no período de 1° de agosto a 30 de novembro de 2000, mas para fins de apuração e divulgação dos dados, a investigação dos domicílios e das pessoas teve como data de referência o dia 1° de agosto de 2000.

4.1.2 Base territorial

O território brasileiro é subdivido em Unidades Político-Administrativas nos diversos níveis de administração (Federal, Estadual e Municipal) e são definidas através de legislação própria. A Base Territorial é um conjunto integrado de mapas, cadastros e bancos de dados, que através de uma metodologia própria (da pesquisa) é construída para dar organização e sustentação espacial às atividades de planejamento operacional, coleta e apuração de dados e divulgação de resultados da pesquisa.

A realização de um levantamento como o Censo Demográfico 2000 representa um desafio para qualquer instituto de estatística, principalmente em um país como o Brasil, que possui 8.514.215,3 km^2 , 27 Unidades da Federação e 5.507 municípios (existentes na data de referência da pesquisa), abrangendo um total de 54.265.618 domicílios.

Neste sentido, para garantir a confiabilidade e precisão dos dados, a coleta é feita seguindo a organização político-administrativa do País. Essa estrutura compreende a União, o Distrito Federal, os Estados e os Municípios, todos autônomos nos termos da Constituição Federal de 5 de outubro de 1988. A divulgação das informações possui essa mesma granularidade espacial.

4.2 Banco Multidimensional de Estatísticas – BME

4.2.1 Aplicação OLAP

O IBGE possui um portal na Internet que oferece à sociedade diferentes ferramentas para acesso aos dados, e uma delas é o Banco Multidimensional de Estatísticas - BME. O BME [3] é um armazém de dados, que tem como objetivo disponibilizar ao público ferramentas voltadas à busca, recuperação e manuseio das informações estatísticas, de forma totalmente desagregada, ou seja, na forma de

microdados[5]. Esse sistema OLAP, que funciona na Internet, é destinado aos pesquisadores que necessitam construir suas próprias informações com os microdados disponíveis e aos profissionais envolvidos em tarefas de planejamento em que seja fundamental o conhecimento da realidade nacional. As consultas realizadas neste sistema são de natureza não previsível e específica (*Ad-hoc*) e executadas sobre uma grande base de dados (própria), com cerca de 1,5 bilhões de registros.

4.2.2 Modelo de dados do Censo Demográfico 2000 no BME

Grande parte das pesquisas produzidas pelo IBGE está disponível no BME, e uma delas é o CD2000, com maior volume de dados (aproximadamente 255 milhões de registros) e maior número de consultas realizadas. O BME é parte integrante do processo de disseminação dos resultados do CD2000.

Para a realização dos experimentos, escolhemos a base de dados da Amostra, que apesar de pouco menor em número de registros em relação à base de dados do Universo, ela é maior em número de variáveis e consultas realizadas, pois reúne mais informações acerca dos domicílios, pessoas e famílias. Essas informações possibilitam aos usuários a realização de consultas investigativas sobre a população, aumentando o número de agregações, o cruzamento de informações, a complexidade de cálculos e o custo de processamento. Essas características foram cruciais para a escolha da base de dados (do Universo ou da Amostra) a ser utilizada nos experimentos desta dissertação.

O modelo lógico de dados do CD2000 especificado no BME segue o modelo estrela para construção de Armazém de Dados, que permite a representação/visualização multidimensional dos dados. Esse modelo é composto por tabelas de fatos e tabelas de dimensão. A base de dados da Amostra do CD2000 no BME é formada por três tabelas de fatos, seguindo a organização de temas da pesquisa: `CD00AMDOMI` (dados de Domicílios), `CD00AMPESS` (dados de Pessoas) e `CD00AMFAMI` (dados de Família), e por 84 tabelas de dimensões, a saber:

- Uma dimensão temporal, denominada `T004`. Essa dimensão refere-se à data de referência da pesquisa;
- Dez dimensões espaciais, denominadas `G000`, `G031`, `G032`, `G033`, `G034`, `G035`, `G036`, `G037`, `G039` e `G042`. Essas dimensões referem-se à base territorial da pesquisa;

[5] Microdados são os dados coletados nos questionários das pesquisas. No BME, cada questionário corresponde a um ou mais registros de informações, sendo a informação mais desagregada possível em uma pesquisa estatística.

- 73 dimensões, denominadas M003, M075, M078, M102, M103, M104, M105, M106, M109, M115, M116, M128, M129, M159, M167, M208, M209, M233, M270, M272, M273, M274, M275, M276, M277, M278, M290, M291, M292, M293, M295, M296, M297, M298, M300, M301, M302, M307, M308, M309, M311, M314, M315, M320, M321, M322, M323, M324, M325, M326, M327, M330, M331, M332, M333, M334, M338, M341, M342, M343, M348, M350, M355, M361, M4210, M4219, M4230, M4239, M4276, M4279, M4300, M4354, M4511.

No Anexo B encontra-se disponível o significado de cada tabela de dimensão.

As tabelas de fatos juntas possuem 31.270.417 registros e formam uma constelação de fatos, pois compartilham algumas tabelas de dimensão. As tabelas de dimensão foram nomeadas seguindo um padrão simples, mas que facilita a identificação das dimensões no modelo: a letra G e um número sequencial para nomear as dimensões espaciais; a letra T e um número sequencial para nomear as dimensões temporais; e a letra M e um número sequencial para as demais dimensões.

A tabela de fatos CD00AMDOMI tem relacionamento 1:N com as tabelas de fatos CD00AMPESS e CD00AMFAMI. A tabela de fatos CD00AMFAMI tem relacionamento 1:N com a tabela de fatos CD00AMPESS e 1:1 com CD00AMDOMI. A tabela de fatos CD00AMPESS tem relacionamento 1:1 com as tabelas de fatos CD00AMFAMI e CD00AMDOMI. A tabela de fatos CD00AMDOMI se relaciona com as tabelas CD00AMPESS e CD00AMFAMI através do atributo CONTROLE. A tabela de fatos CD00AMPESS se relaciona com a tabela CD00AMFAMI através dos atributos CONTROLE e V0404, e vice-versa. A Figura 9 mostra o relacionamento entre essas tabelas.

Todas as tabelas de fatos se relacionam com todas as dimensões espaciais e temporal. As tabelas de fatos CD00AMDOMI, CD00AMFAMI e CD00AMPESS têm relacionamento 1:1 com cada tabela de dimensão espacial e temporal. Cada dimensão espacial e temporal tem relacionamento 1:N com cada tabela de fatos. A Figura 10 mostra o relacionamento entre as tabelas de fatos e as dimensões espacial e temporal, bem como a cardinalidade de cada tabela. Cada tabela de fatos se relaciona com todas as dimensões espaciais e temporal.

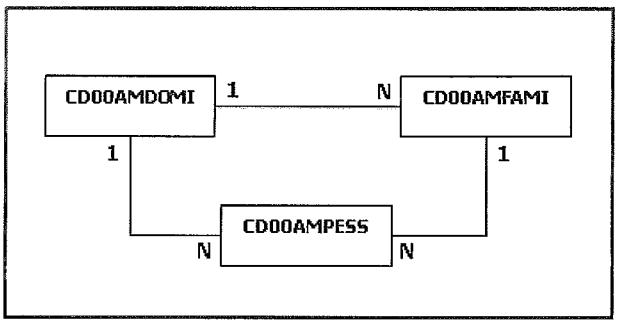


Figura 9: Relacionamento entre as tabelas de fatos CD00AMDOMI, CD00AMFAMI e CD00AMPESS.

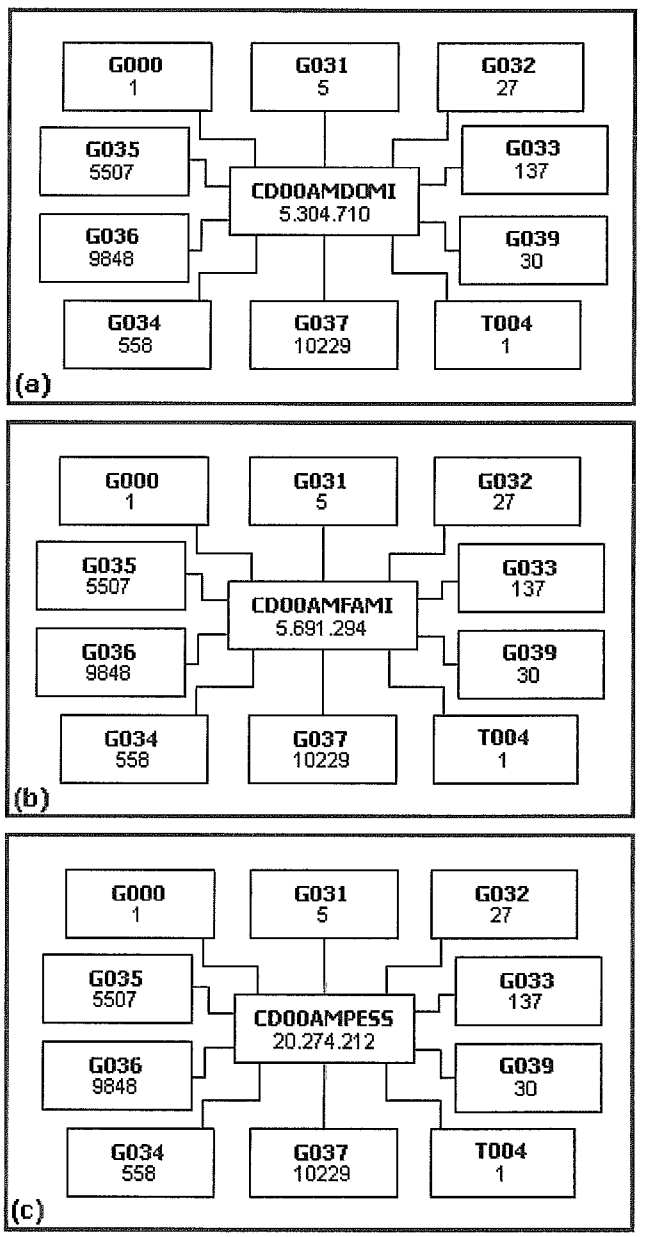


Figura 10: Relacionamento entre tabelas de fatos e dimensões espacial e temporal. (a) relacionamento entre a tabela de fatos CD00AMDOMI e dimensões; (b) relacionamento entre a tabela de fatos CD00AMFAMI e dimensões; (c) relacionamento entre a tabela de fatos CD00AMPESS e dimensões.

As tabelas de dimensão espaciais relacionam-se entre si para definirem a hierarquia geográfica descrita na seção 4.1.2. O relacionamento entre as dimensões espaciais do maior nível de agregação para o menor ocorre da seguinte forma: a dimensão que representa o maior nível de agregação (País) tem relacionamento 1:N com a dimensão que representa um nível de agregação abaixo (Regiões Geográficas); essa dimensão tem relacionamento 1:N com a dimensão que representa um nível de agregação abaixo (Unidades da Federação), e assim sucessivamente, até o relacionamento 1:N entre as dimensões que representam Distritos e Subdistritos (que é o menor nível de agregação).

O relacionamento entre as dimensões espaciais do menor nível de agregação para o maior ocorre da seguinte forma: a dimensão que representa Subdistritos tem relacionamento 1:1 com a dimensão que representa um nível de agregação acima (Distritos); essa dimensão tem relacionamento 1:1 com a dimensão que representa um nível acima (Municípios), e assim sucessivamente, até o relacionamento 1:1 entre Regiões Geográficas e País.

Embora exista relacionamento entre as tabelas de fatos e todas as tabelas de dimensão espaciais e temporal, não existe relacionamento entre as tabelas de fatos e todas as demais dimensões. Algumas dimensões se relacionam apenas com a tabela de fatos CD00AMDOMI; outras dimensões se relacionam apenas com a tabela de fatos CD00AMFAMI; e as dimensões que não se relacionam com nenhuma dessas tabelas de fatos, se relacionam então com a tabela de fatos CD00AMPESS.

Além das dimensões espacial e temporal, as tuplas de CD00AMDOMI referenciam as tuplas de M003, M075, M078, M102, M103, M104, M105, M106, M109, M115, M116, M128, M129, M208, M209, M233, M270, M272, M273, M274, M275, M276, M277, M278. Cada uma dessas tabelas de dimensão tem relacionamento 1:N com a tabela CD00AMDOMI. A Figura 11 ilustra o relacionamento entre a tabela de fatos CD00AMDOMI e as 24 dimensões, com exceção das dimensões espaciais e temporal.

As tuplas de CD00AMPESS referenciam as tuplas de M159, M167, M298, M300, M301, M302, M307, M308, M309, M311, M314, M315, M320, M321, M322, M323, M324, M325, M326, M327, M330, M331, M332, M333, M334, M338, M341, M342, M343, M348, M350, M355, M361, M4210, M4219, M4230, M4239, M4276, M4279, M4300, M4354, M4511, além das dimensões espacial e temporal. Cada uma dessas tabelas de dimensão tem relacionamento 1:N com

a tabela CD00AMPESS. A Figura 12 ilustra o relacionamento entre a tabela de fatos CD00AMPESS e as 42 dimensões, com exceção das dimensões espaciais e temporal.

E as tuplas de CD00AMFAMI referenciam as tuplas de M290, M291, M292, M293, M295, M296, M297, além das dimensões espacial e temporal. Cada uma dessas tabelas de dimensão tem relacionamento 1:N com a tabela CD00AMFAMI. A Figura 13 ilustra o relacionamento entre a tabela de fatos CD00AMFAMI e as 7 dimensões, com exceção das dimensões espaciais e temporal.

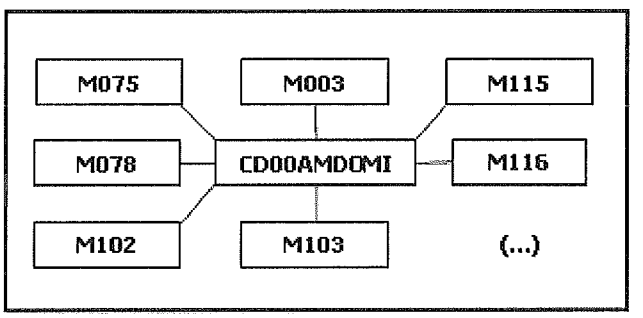


Figura 11: Relacionamento entre a tabela de fato CD00AMDOMI e suas dimensões.

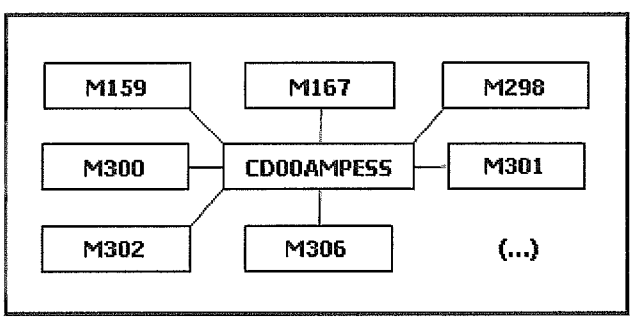


Figura 12: Relacionamento entre a tabela de fato CD00AMPESS e suas dimensões.

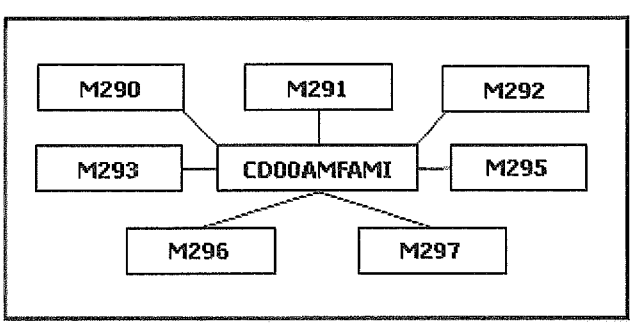


Figura 13: Relacionamento entre a tabela de fato CD00AMFAMI e suas dimensões.

Durante a coleta de dados no CD2000, todos os domicílios recebem um número único que os identificam no país. Ao fornecer os microdados da pesquisa, o IBGE desidentifica todos os domicílios a fim de garantir o sigilo da informação prestada, prevista em lei[6] [20]. A desidentificação é feita substituindo o identificador do domicílio por um número sequencial pseudo-aleatório gerado por um algoritmo específico, com ciclo maior que o número de registros. Esse número sequencial é armazenado no atributo CONTROLE, chave primária da tabela de fatos CD00AMDOMI. Esse atributo compõe as chaves primárias de CD00AMFAMI e CD00AMPESS junto com o número de ordem da família no domicílio (V0404) e o número de ordem da pessoa na família (V0400). A Tabela 2 apresenta as chaves primárias das tabelas de fato. A chave primária de cada tabela de dimensão é o atributo CODIGO, presente em todas essas tabelas.

Tabela 2: Chaves primárias das tabelas de fatos

TABELA DE FATOS	CHAVE PRIMÁRIA
CD00AMDOMI	CONTROLE
CD00AMPESS	CONTROLE, V0404, V0400
CD00AMFAMI	CONTROLE, V0404

A chave de junção entre CD00AMDOMI e as demais tabelas de fato (CD00AMFAMI e CD00AMPESS) é formada pelo atributo CONTROLE. Entre CD00AMPESS e CD00AMFAMI, a chave de junção é composta pelos atributos CONTROLE e V0404.

Conforme descrito acima, os dados de domicílios estão armazenados na tabela de fatos CD00AMDOMI, que possui 65 atributos. Além da identificação do domicílio, da localização geográfica e o peso do domicílio, esses atributos contêm informações sobre bens duráveis, saneamento básico, características físicas do domicílio, rendimento domiciliar e outras características de condições de moradia. A tabela de fatos CD00AMPESS, que armazena dados de pessoas, possui 107 atributos. Nesses atributos estão armazenadas informações sobre fecundidade, instrução, migração, nupcialidade, previdência, rendimentos e demais características físicas da pessoa, além da identificação da pessoa, da família, do domicílio, da localização geográfica e peso da pessoa. A tabela de fatos CD00AMFAMI armazena dados de família e possui 28 atributos,

6 Lei nº 5.534 – Obrigatoriedade de prestação de informações estatísticas (14.11.68)
"A lei que rege a obrigatoriedade de prestação de informações estatísticas informa o cidadão brasileiro acerca de sua responsabilidade de ajudar o país com segurança, sabendo que toda informação fornecida terá fins exclusivamente estatísticos. Através da Lei nº 5.534 de 14 de novembro de 1968, o cidadão tem garantido seu direito de sigilo estatístico e seu dever de prestar informações estatísticas ao IBGE."

com informações sobre o número de componentes da família, tipo da família, rendimento familiar, além da identificação da pessoa, do domicílio, da localização geográfica e peso da família. Todas os atributos das tabelas de fatos, sem exceção, são do tipo numérico. A Figura 14 apresenta os atributos das tabelas de fatos `CD00AMDOMI`, `CD00AMFAMI` e `CD00AMPESS`. No Anexo A encontram-se disponíveis as descrições dos atributos das tabelas de fatos.

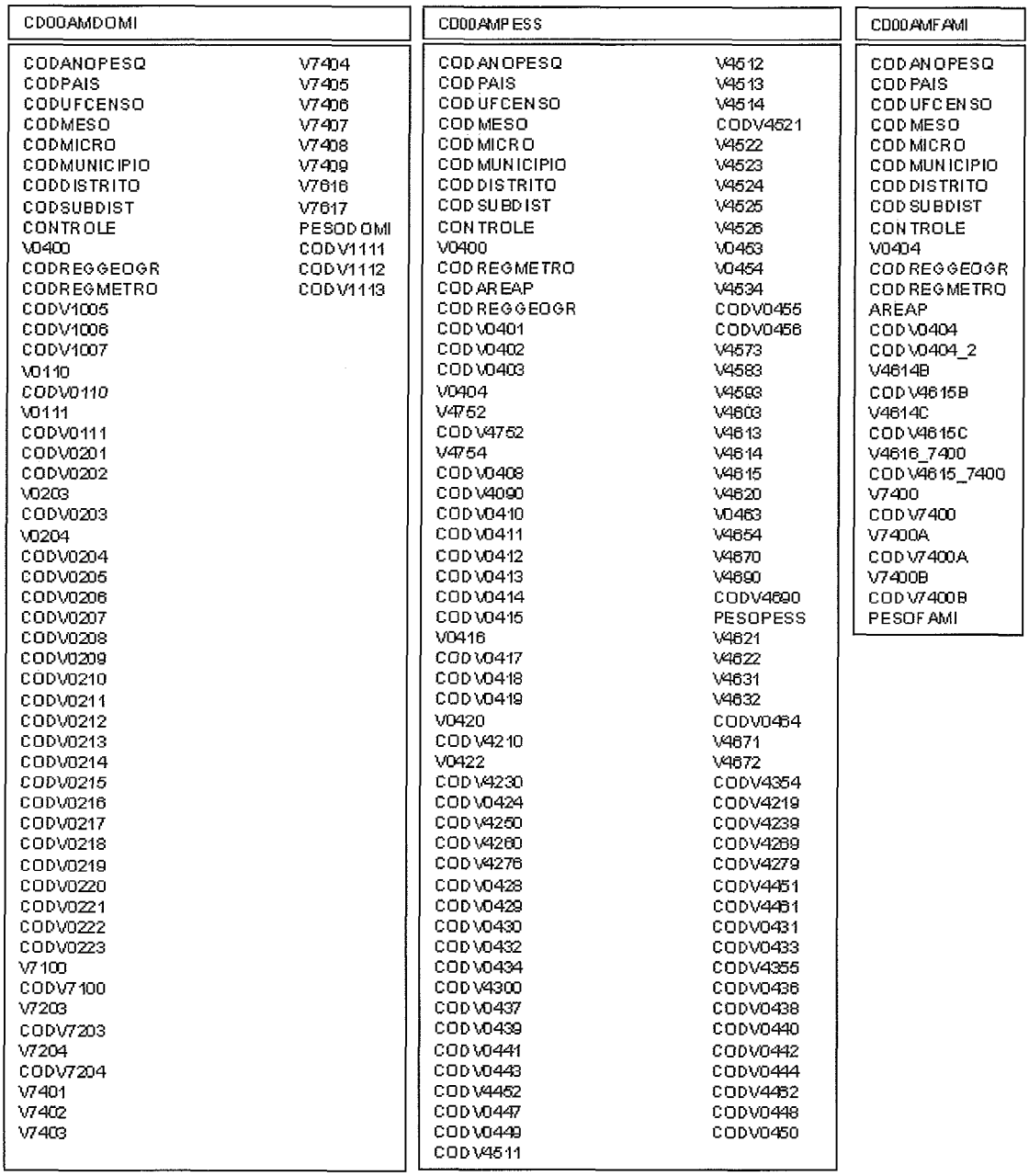


Figura 14: Atributos das tabelas de fatos CD00AMDOMI, CD00AMFAMI e CD00AMPESS

Todas as tabelas de dimensão (inclusive espaciais e temporais) possuem três atributos: `CODIGO`, `DENOMINACAO` e `METADADOS`. Esses atributos armazenam o código, a

descrição e os metadados (informações relevantes aos fatos armazenados nas tabelas de fatos). No Anexo B encontram-se disponíveis as descrições dos atributos das tabelas de dimensão.

As tabelas físicas que armazenam os dados da Amostra do CD2000 no BME representam as tabelas de fatos e de dimensão do modelo lógico descrito na seção anterior. Todas as tabelas possuem índices primários, que representam as chaves primárias, e os relacionamentos diversos entre as tabelas (de fatos e de dimensão) são feitas através de chaves estrangeiras (vide Anexo C). Sobre os atributos `CONTROLE` e `V0404`, que formam a chave de junção entre as tabelas de fatos `CD00AMFAMI` e `CD00AMPESS`, foram criados índices (de junção) compostos do tipo *B-tree*. Na tabela `CD00AMDOMI` não foi necessário a criação de índice porque o atributo `CONTROLE` já possui um índice primário (chave primária).

Nos demais atributos, os índices secundários foram criados de acordo com o tipo do atributo. Sobre os atributos que representam as variáveis medidas (aquelas que quantificam um domínio) foram criados índices do tipo *B-tree*; sobre os atributos que representam as variáveis observadas (aquelas que classificam um domínio) foram criados índices do tipo *bitmap*.

4.2.3 Projeto de fragmentação do Censo Demográfico 2000 no BME

O projeto de distribuição da base de dados do CD2000 no BME é particionado e não replicado, e cada disco do servidor de banco de dados é um local para o armazenamento de diferentes fragmentos das relações da base. Neste servidor está instalado um SGBD paralelo comercial para gerenciamento da base de dados do BME. Por estar instalado em um computador que possui multiprocessadores, o SGBD tira proveito dessa arquitetura para processar as consultas em paralelo, já que o mesmo oferece essa funcionalidade.

As tabelas de fatos do CD2000 armazenam o maior número de registros, e com exceção da dimensão G035, as tabelas de dimensão são bem pequenas, em número de linhas. Por esse motivo, a fragmentação da base de dados do CD2000 no BME é realizada apenas sobre as tabelas de fatos, porque além de possuir alta cardinalidade, as consultas realizadas no BME são feitas sempre sobre essas tabelas.

A estratégia de fragmentação utilizada sobre as tabelas foi a fragmentação horizontal primária. Para essa fragmentação, foram utilizadas informações acerca da

cardinalidade das tabelas, de informações qualitativas (que consiste nos predicados usados nas consultas dos usuários) e informações quantitativas (que consiste no número de linhas retornadas por uma consulta e freqüência das consultas). As consultas realizadas sobre a base de dados do BME são de natureza não previsível, no entanto, por ser um banco de dados geoestatísticos, grande parte das consultas possuem predicados relacionados à espacialidade da informação.

Considerando as relações `CD00AMDOMI`, `CD00AMFAMI` e `CD00AMPESS` descritas na seção anterior, alguns predicados podem ser definidos:

p_1: `CODUFCENSO = 11`
p_2: `CODUFCENSO = 29`
p_3: `CODUFCENSO = 33`
p_4: `CODUFCENSO = 42`
p_5: `CODUFCENSO = 50`
p_6: `CODREGGEOGR = 1`
p_7: `CODREGGEOGR = 2`
p_8: `CODREGGEOGR = 3`
p_9: `CODREGGEOGR = 4`
p_{10}: `CODREGGEOGR = 5`

Os predicados acima apresentados são apenas uma amostra representativa e oriundos de uma análise sobre os resultados divulgados na publicação do Censo Demográfico 2000 [15], dentre outras, divulgadas pelo IBGE. Parte das tabelas de resultados dessas publicações apresenta as informações segundo uma unidade territorial (Brasil, Unidade da Federação, Grandes Regiões e Municípios). Em geral, consultas com esses predicados ocorrem muito freqüentemente no BME.

Sobre os predicados definidos, foi realizada uma análise a respeito do número de linhas lidas/retornadas quando do uso desses predicados a fim de se determinar o tamanho dos fragmentos a serem gerados. O número de linhas por Grandes regiões e Unidades da Federação é apresentado na Tabela 3.

Para cada tabela de fatos foram gerados cinco fragmentos, com tamanho aproximado de número de linhas, agrupadas segundo a Unidade da Federação. As Unidades da Federação foram agrupadas segundo a Região Geográfica a qual fazia parte, com exceção do estado de São Paulo, que sozinho possui quase o mesmo número de linhas de outros estados juntos.

Tabela 3: Número de linhas das tabelas de fatos por Grandes Regiões e Unidades da Federação (em número de registros)

		Tabela de fatos		
Grandes Regiões	**Unidades da Federação**	**CD00AMDOMI**	**CD00AMFAMI**	**CD00AMPESS**
1 - Norte	11 - Rondônia	43.293	45.856	172.073
	12 - Acre	16.818	18.113	71.063
	13 - Amazonas	63.970	73.677	314.758
	14 - Roraima	9.857	10.576	41.639
	15 - Pará	145.992	166.831	691.394
	16 - Amapá	11.821	13.338	55.391
	17 - Tocantins	43.043	45.832	175.904
2 - Nordeste	21 - Maranhão	150.441	166.793	703.621
	22 - Piauí	94.534	103.284	405.936
	23 - Ceará	201.143	219.843	866.347
	24 - Rio Grande do Norte	92.673	102.556	390.126
	25 - Paraíba	117.577	128.391	487.848
	26 - Pernambuco	225.649	247.081	935.536
	27 - Alagoas	77.896	85.836	348.429
	28 - Sergipe	55.161	60.248	230.984
	29 - Bahia	378.907	414.688	1.598.126
3 - Sudeste	31 - Minas Gerais	615.101	656.095	2.347.758
	32 - Espírito Santo	98.820	105.145	369.666
	33 - Rio de Janeiro	442.976	472.432	1.511.640
	35 - São Paulo	1.137.154	1.200.093	4.038.217
4 - Sul	41 - Paraná	336.151	355.118	1.218.361
	42 - Santa Catarina	193.633	204.506	693.703
	43 - Rio Grande do Sul	365.827	386.340	1.209.631
5 - Centro-Oeste	50 - Mato Grosso do Sul	69.401	74.056	251.403
	51 - Mato Grosso	86.946	91.846	326.022
	52 - Goiás	175.132	184.052	617.948
	53 - Distrito Federal	54.795	58.668	200.888

A decomposição da relação CD00AMDOMI em fragmentos horizontais $CD00AMDOMI_1$, $CD00AMDOMI_2$, $CD00AMDOMI_3$, $CD00AMDOMI_4$ e $CD00AMDOMI_5$ é definida da seguinte forma:

$$CD00AMDOMI_1 = \sigma_{11 \leq CODUFCENSO \leq 23}(\mathbf{CD00AMDOMI})$$

$$CD00AMDOMI_2 = \sigma_{24 \leq CODUFCENSO \leq 29}(\mathbf{CD00AMDOMI})$$

$$CD00AMDOMI_3 = \sigma_{31 \leq CODUFCENSO \leq 33}(\mathbf{CD00AMDOMI})$$

$$CD00AMDOMI_4 = \sigma_{41 \leq CODUFCENSO \leq 53}(\mathbf{CD00AMDOMI})$$

$$CD00AMDOMI_5 = \sigma_{CODUFCENSO = 35}(\mathbf{CD00AMDOMI})$$

A decomposição da relação CD00AMFAMI em fragmentos horizontais $CD00AMFAMI_1$, $CD00AMFAMI_2$, $CD00AMFAMI_3$, $CD00AMFAMI_4$ e $CD00AMFAMI_5$ é definida da seguinte forma:

$$CD00AMFAMI_1 = \sigma_{11 \leq CODUFCENSO \leq 23}(\mathbf{CD00AMFAMI})$$

$$CD00AMFAMI_2 = \sigma_{24 \leq CODUFCENSO \leq 29}(\mathbf{CD00AMFAMI})$$

$$CD00AMFAMI_3 = \sigma_{31 \leq CODUFCENSO \leq 33}(\mathbf{CD00AMFAMI})$$

$$CD00AMFAMI_4 = \sigma_{41 \leq CODUFCENSO \leq 53}(\mathbf{CD00AMFAMI})$$

$$CD00AMFAMI_5 = \sigma_{CODUFCENSO = 35}(\mathbf{CD00AMFAMI})$$

A decomposição da relação CD00AMPESS em fragmentos horizontais $CD00AMPESS_1$, $CD00AMPESS_2$, $CD00AMPESS_3$, $CD00AMPESS_4$ e $CD00AMPESS_5$ é definida da seguinte forma:

$$CD00AMPESS_1 = \sigma_{11 \leq CODUFCENSO \leq 23}\ (\mathbf{CD00AMPESS})$$

$$CD00AMPESS_2 = \sigma_{24 \leq CODUFCENSO \leq 29}\ (\mathbf{CD00AMPESS})$$

$$CD00AMPESS_3 = \sigma_{31 \leq CODUFCENSO \leq 33}\ (\mathbf{CD00AMPESS})$$

$$CD00AMPESS_4 = \sigma_{41 \leq CODUFCENSO \leq 53}\ (\mathbf{CD00AMPESS})$$

$$CD00AMPESS_5 = \sigma_{CODUFCENSO = 35}\ (\mathbf{CD00AMPESS})$$

Após a decomposição das relações, os fragmentos gerados foram verificados em relação aos critérios de correção (completeza, reconstrução e disjunção), a fim de assegurar a consistência da base de dados do CD2000 e afirmamos que:

$$CD00AMDOMI = CD00AMDOMI_1 \cup CD00AMDOMI_2 \cup CD00AMDOMI_3 \cup CD00AMDOMI_4 \cup CD00AMDOMI_5$$

$$CD00AMFAMI = CD00AMFAMI_1 \cup CD00AMFAMI_2 \cup CD00AMFAMI_3 \cup CD00AMFAMI_4 \cup CD00AMFAMI_5$$

$$CD00AMPESS = CD00AMPESS_1 \cup CD00AMPESS_2 \cup CD00AMPESS_3 \cup CD00AMPESS_4 \cup CD00AMPESS_5$$

Uma vez definidos, os fragmentos foram distribuídos em diferentes discos do computador. Embora esses fragmentos estejam presentes em um mesmo local (mas em discos separados), esse tipo de organização proporciona ganho de desempenho durante o acesso aos dados no disco. Se uma consulta for realizada tendo como predicado duas Unidades da Federação contidas em partições diferentes, a leitura desses fragmentos será feita em paralelo, ao contrário do que seria se as linhas estivessem armazenadas em um único fragmento.

4.2.4 Consultas *Ad-hoc* sobre o Censo Demográfico 2000

As consultas realizadas no BME são de natureza não previsível e específicas (*Ad-hoc*), executadas consecutivamente como uma espécie de refinamento da consulta inicial; consultas diferentes que se complementam também podem ser executadas consecutivamente. Muitas vezes o resultado obtido em uma determinada consulta direciona o usuário sobre que tipos de consultas podem ser realizadas a fim de alcançar o resultado esperado. Essas características são típicas de consultas feitas por um usuário em um sistema OLAP.

No BME, essas consultas de análise são realizadas em tempo real, ou seja, são feitas diretamente sobre os dados (sem a intervenção de pessoas), em qualquer momento e de qualquer lugar (o sistema está sempre disponível na Internet). A sua interface gráfica permite o usuário escolher livremente as variáveis que irão compor a consulta, o espaço geográfico e o período temporal de referência das informações. O processamento da consulta pode resultar, conforme opção do usuário, na apresentação de uma tabela com os resultados já agregados ou num arquivo no formato "csv - comma separated value", com os microdados selecionados. O cruzamento das variáveis pode ser reordenado para obter-se nova apresentação dos dados.

No BME, as pesquisas são organizadas e apresentadas em uma árvore, denominada árvore de informações, representadas por uma figura de "pasta". Dentro de cada pasta, encontra-se disponível as variáveis das pesquisas. O usuário navega nesta árvore para selecionar as variáveis de uma pesquisa que irão compor a sua consulta. Após a escolha das variáveis, o usuário define filtros temporais e/ou espaciais e demais filtros (nas variáveis). À medida que o usuário define a sua consulta de forma gráfica, o BME gera o código SQL (*Structured Query Language*) que será realizado diretamente sobre a base de dados. A geração do SQL é feita de forma transparente e o usuário não tem acesso ao mesmo. O Administrador da aplicação BME tem acesso ao SQL apenas para visualização. Como os dados do BME estão armazenados em um SGBD paralelo comercial, as consultas possuem funções específicas desse SGBD. As consultas realizadas no BME são executadas em dois passos. No primeiro passo, resolve-se o cálculo da informação e no segundo passo, a codificação da informação, ou seja, decodifica as categorias obtidas na agregação do primeiro passo. A Figura 15 apresenta o SQL de uma consulta definida por um usuário no BME. Os SQL utilizados nos experimentos desta dissertação foram gerados pelo BME.

PASSO 1:

```
SELECT CD00AMPESS.CODUFCENSO GEOGRAFIA_COD , 9032 GEOGRAFIA_DIM_COD ,
CD00AMPESS.CODANOPESQ TEMPO_COD , 8005 TEMPO_DIM_COD , NVL(CD00AMPESS.CODV0401, -1) COL1
SUM(CD00AMPESS.V4752 * CD00AMPESS.PESO) / SUM(CD00AMPESS.PESO) COL2 , SUM(CD00AMPESS.PESO)
frequencia , count(*) contagem FROM IBGE.CD00AMPESS WHERE (CD00AMPESS.CODUFCENSO = '12') AND
CD00AMPESS.CODUFCENSO in (select Codigo from G032 where IND_EXIBICAO = 'S' ) AND (CD00AMPESS.V4752
BETWEEN 7 AND 14) GROUP BY CD00AMPESS.CODUFCENSO , CD00AMPESS.CODANOPESQ , NVL
(CD00AMPESS.CODV0401, -1)
```

PASSO 2:

```
SELECT BMM_TMP_241657.GEOGRAFIA_COD , BMM_TMP_241657.GEOGRAFIA_DIM_COD , G032.Denominacao
geografia_desc , BMM_TMP_241657.TEMPO_COD , BMM_TMP_241657.TEMPO_DIM_COD , T005.Denominacao
tempo_desc , tab1.Denominacao COL1 , tab1.Ordem COL1_ord , tab1.Codigo COL1_cod , BMM_TMP_241657.COL2 ,
BMM_TMP_241657.FREQUENCIA , BMM_TMP_241657.CONTAGEM FROM G032 , T005 , M300 tab1 ,
BMM_TMP_241657 WHERE (BMM_TMP_241657.GEOGRAFIA_COD = '12') AND
BMM_TMP_241657.GEOGRAFIA_DIM_COD = 9032 AND (BMM_TMP_241657.TEMPO_COD = '2000') AND
BMM_TMP_241657.TEMPO_DIM_COD = 8005 AND BMM_TMP_241657.COL1 = tab1.Codigo AND
BMM_TMP_241657.TEMPO_COD = T005.Codigo AND BMM_TMP_241657.GEOGRAFIA_COD = G032.Codigo
```

Figura 15: SQL de uma consulta definida por um usuário no BME

Todas as consultas realizadas no BME realizam junções com no mínimo uma tabela de fatos e duas tabelas de dimensões: uma dimensão temporal e uma dimensão espacial, e realizam no mínimo duas agregações. No caso de pesquisas amostrais, como o CD2000, o BME realiza automaticamente a expansão dos dados (vide seção 4.1).

A base de dados do BME possui aproximadamente 1,5 bilhão de registros de informações sobre o Brasil, distribuídos entre as diversas pesquisas realizadas pelo IBGE e que estão disponíveis nessa base. Desse número, o CD2000 possui aproximadamente 255 milhões de registros. Sobre esse grande acervo de dados, parte das consultas realizadas pelos usuários do BME é feita *on-line*, com resultados sendo exibidos no momento da consulta. No entanto, algumas consultas, principalmente as realizadas sobre a base do CD2000, não podem ser realizadas imediatamente e são agendadas para serem executadas em outra ocasião. Essas consultas agendadas, geralmente envolvem muitas junções, agregações e tem muitas variáveis selecionadas, o que gera um custo elevado no processamento e no tempo despendido para o cálculo da informação. O processamento de uma consulta desse porte durante um momento de pico de utilização da aplicação pode acarretar a degradação de tempo no processamento de consultas menos complexas, afetando muitos usuários. Optou-se então, pelo agendamento de consultas complexas e custosas como solução para o problema descrito.

No BME o número de usuários conectados e realizando consultas simultaneamente é sazonal, ou seja, varia de acordo com as diversas épocas de divulgação de pesquisas. À medida em que se aproxima a data de publicação de uma pesquisa, a utilização da aplicação aumenta, atingindo o seu número máximo no dia da divulgação. Outro fator que implica um grande número de consultas concorrentes é a

época em que ocorrem Treinamentos de Ferramentas Digitais do IBGE, em que se tem no mínimo 20 usuários conectados, realizando consultas, por turma. A carga de dados das pesquisas para divulgação e a conferência destes pelos diversos departamentos também intensificam, e muito, o uso do BME. Ou seja, em determinados momentos, o uso da aplicação é baixo, em outros, muito alto. Isso nos leva a ter consultas sendo executadas isoladamente (sem concorrência), e consultas sendo executadas simultaneamente, ou seja, concorrentemente, elevando a carga de processamento dessas consultas.

Atualmente, o BME possui 1.465 usuários cadastrados, e entre Janeiro de 2006 e Outubro de 2007, 361 usuários realizaram pelo menos uma consulta, totalizando 46.353 consultas nesse período. Desse número, 13.143 foram feitas na base do CD2000, sendo 10.813 realizadas sobre os dados da Amostra.

Analisando os registros de utilização do BME entre Janeiro de 2006 e Outubro de 2007, observamos o máximo de 49 consultas concorrentes. A média de consultas concorrentes é 3.7, com desvio padrão igual a 3.8 e mediana igual a 2. Em relação ao número de consultas realizadas por um usuário em um dia, observamos o número máximo de 188 consultas consecutivas. A média de consultas consecutivas por usuário é 8, com desvio padrão igual a 12.6 e mediana igual a 4.

É importante salientar que o fato de um usuário estar conectado no BME não quer dizer que ele execute alguma consulta, isto é, o usuário pode estar apenas navegando pela meta-informação. E o usuário não pode executar mais de uma consulta ao mesmo tempo, ou seja, quando ele envia uma consulta para o banco, ele permanece aguardando o resultado. Após o recebimento do resultado de uma consulta, ele pode executar outra. Portanto, em nossa análise, levamos em conta o número de consultas sendo processadas concorrentemente e não o número de usuários conectados simultaneamente.

Também deve ser observado que as consultas não são processadas concorrentemente do início ao fim, ou seja, as consultas podem estar concorrentes durante apenas um intervalo de tempo (enquanto uma está iniciando o processamento a outra está finalizando). Não foi possível identificar o tempo de interseção do processamento das consultas.

Nos registros de utilização do BME não há informações sobre que consultas foram executadas, constam apenas informações sobre quais pesquisas as consultas foram realizadas. O administrador do sistema tem acesso apenas ao conteúdo das consultas (variáveis selecionadas e filtros aplicados) que foram agendadas para

serem executadas em outra ocasião. Por esse motivo, as consultas dos experimentos desta dissertação foram escolhidas através da nossa experiência com o manuseio dos dados, da aplicação BME e das consultas agendadas; do nosso contato com as pessoas que utilizam a aplicação e com as pessoas durante os treinamentos de uso do BME; e da análise de algumas publicações de resultados do CD2000 para identificar possíveis consultas realizadas no BME.

Para a realização dos experimentos desta dissertação foram selecionadas 14 consultas de alto custo, tipicamente realizadas sobre a base de dados do CD2000 no BME. Essas consultas possuem diferentes níveis de complexidade, que foram definidos levando-se em conta o número de registros lidos, o número de agregações, o número de junções e os fragmentos lidos. A seguir, apresentamos e descrevemos as consultas selecionadas, Q1 a Q14:

Q1: Número de domicílios segundo a situação em relação à sua localização (urbano e rural) no Brasil.

Essa consulta acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios segundo a situação em relação à sua localização (urbano ou rural). No segundo passo, faz uma junção com as dimensões G000, T004 e M208, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado, o que a faz utilizar todas as tuplas da tabela. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q2: Número de pessoas por classes de idade em anos no Brasil.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandida e simples, de pessoas para classes de idade da população. No segundo passo, faz uma junção com as dimensões G000, T004 e M302, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado, o que a faz utilizar todas as tuplas da tabela. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q3: Média da idade em anos das pessoas no Brasil.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a média e freqüência, expandidas e simples, da idade em anos das pessoas. No segundo passo, faz uma junção com as dimensões G000 e T004, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado o que a faz utilizar todas as

tuplas da tabela. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q4: Total de filhos tidos pelas pessoas no Brasil, por classes de número de filhos.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a soma e freqüência, expandidas e simples, do número de filhos tidos por classes de número de filhos. No segundo passo, faz uma junção com as dimensões G000, T004 e M348, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado o que a faz utilizar todas as tuplas da tabela. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q5: Número de pessoas no Brasil com determinadas religiões.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandida e simples, de pessoas por tipo de religião. No segundo passo, faz uma junção com as dimensões G000, T004 , M307, decodificando as categorias obtidas na agregação. Essa consulta possui um predicado seletivo, recuperando 83,95% das tuplas. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q6: Média da idade em anos das pessoas no Brasil com idade entre 20 e 40 anos.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência e a média, expandidas e simples, de pessoas entre 20 e 40 anos de idade. No segundo passo, faz uma junção com as dimensões G000 e T004 decodificando as categorias obtidas na agregação. Essa consulta possui um predicado seletivo, recuperando 33,17% das tuplas. Todos os fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q7: Média de anos de estudo das pessoas, por sexo, no município do Rio de Janeiro.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência e a média, expandidas e simples, de anos de estudo das pessoas, por sexo, no município do Rio de Janeiro. No segundo passo, faz junção com as dimensões G035, T004 e M300, decodificando as categorias obtidas na agregação. Essa consulta possui um predicado espacial bem seletivo, recuperando 2,92% das tuplas. Apenas um fragmento da tabela de fatos é lido durante a execução dessa consulta.

Q8: Média de anos de estudo das pessoas, por sexo, nos municípios do Rio de Janeiro e Natal.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência e a média, expandidas e simples, de anos de estudo das pessoas, por sexo, no município do Rio de Janeiro. No segundo passo, faz a junção com as dimensões G035, T004 e M300, decodificando as categorias obtidas na agregação. Essa consulta possui dois predicados espaciais bem seletivos, recuperando 3,28% das tuplas. Apenas dois fragmentos da tabela de fatos são lidos durante a execução dessa consulta.

Q9: Número de pessoas no Brasil, por sexo, em zonas rurais e urbanas.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandidas e simples, do número de pessoas por sexo; e acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios em zonas rurais e urbanas. É realizada uma junção entre as tabelas de fatos CD00AMPESS e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de fatos e as dimensões G000, T004, M208 e M300, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado o que a faz utilizar todas as tuplas da tabela. Todos os fragmentos das duas tabelas de fatos são lidos durante a execução dessa consulta.

Q10: Média de anos de estudo das pessoas, por sexo, nas zonas rurais e urbanas do município do Rio de Janeiro.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência e média, expandidas e simples, dos anos de estudos das pessoas por sexo; e acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios em zonas rurais e urbanas. É realizada uma junção entre as tabelas de fatos CD00AMPESS e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de fatos e as dimensões G035, T004, M208 e M300, decodificando as categorias obtidas na agregação. Essa consulta possui dois predicados bem seletivos: um de seleção e um espacial, recuperando 2,32% das tuplas. Apenas um fragmento de cada tabela de fatos é lido durante a execução dessa consulta.

Q11: Média de anos de estudo das pessoas, por sexo, nas zonas rurais e urbanas dos municípios do Rio de Janeiro e Natal.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência e média, expandidas e simples, dos anos de estudos das pessoas por sexo; e acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios em zonas rurais e urbanas. É realizada uma junção entre as tabelas de fatos CD00AMPESS e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de

fatos e as dimensões G035, T004, M208 e M300, decodificando as categorias obtidas na agregação. Essa consulta possui dois predicados seletivos: um de seleção e dois espaciais, recuperando 2,56% das tuplas. Apenas dois fragmentos das duas tabelas de fatos são lidos durante a execução dessa consulta.

Q12: Bolsa-Escola - Característica de pobreza dos domicílios particulares permanentes com crianças em fase escolar em cada Unidade da Federação.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandidas e simples, de crianças em fase escolar (entre 7 e 14 anos); acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios particulares permanentes; e acessa a tabela de fatos CD00AMFAMI para calcular a freqüência, expandida e simples, de famílias com rendimento mensal per-capita inferior a R$121,00. É realizada uma junção entre as tabelas de fatos CD00AMPESS, CD00AMFAMI e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de fatos e as dimensões G032, T004 e M270, decodificando as categorias obtidas na agregação. Essa consulta possui dois predicados bem seletivos: de seleção e espacial, recuperando 6,87% das tuplas. Todos os fragmentos das tabelas de fatos são lidos durante a execução dessa consulta.

Q13: Número de pessoas que sabem e não sabem ler, nas zonas urbanas e rurais em cada Unidade da Federação.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandidas e simples, de pessoas que sabem e não sabem ler; e acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios em zonas rurais e urbanas. É realizada uma junção entre as tabelas de fatos CD00AMPESS e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de fatos e as dimensões G032, T004 e M320, decodificando as categorias obtidas na agregação. Essa consulta não possui nenhum predicado o que a faz utilizar todas as tuplas da tabela. Todos os fragmentos das duas tabelas de fatos são lidos durante a execução dessa consulta.

Q14: Deficientes - Calcular o número de famílias, por tipo de família, que tenham algum de seus membros com deficiência mental ou de ouvir, em zonas rurais e urbanas em cada Unidade da Federação.

Essa consulta acessa a tabela de fatos CD00AMPESS para calcular a freqüência, expandidas e simples, de pessoas com deficiência física; acessa a tabela de fatos CD00AMDOMI para calcular a freqüência, expandida e simples, de domicílios em

zonas rurais e urbanas; e acessa a tabela de fatos CD00AMFAMI para calcular a freqüência, expandida e simples, de famílias por tipo de famílias. É realizada uma junção entre as tabelas de fato CD00AMPESS, CD00AMFAMI e CD00AMDOMI. No segundo passo, faz a junção entre as tabelas de fatos com as dimensões G032, T004 e M208, M295, M308 e M361, decodificando as categorias obtidas na agregação. Essa consulta possui dois predicados bem seletivos: de seleção e espacial, recuperando 1,13% das tuplas. Entre os predicados é utilizado o operador OR (disjunção). Todos os fragmentos das tabelas de fatos são lidos durante a execução dessa consulta.

Essas informações acerca das consultas realizadas no BME, como fragmentos lidos, número de linhas retornadas, número de consultas isoladas e concorrentes, número de usuários conectados, foram essenciais para a definição dos experimentos realizados nesta dissertação.

4.3 Modelo de dados e Projeto de fragmentação do Censo Demográfico 2000 no ParGRES

Para a realização dos experimentos desta dissertação utilizamos a metodologia proposta no Capítulo 3 para definir o modelo de dados e o projeto de fragmentação do CD2000 no ParGRES. Apesar de não ser necessária qualquer alteração no modelo da base durante a migração para o ambiente distribuído, foram necessárias algumas modificações no projeto físico do CD2000, impostas por um fator limitador em nossos experimentos: o espaço físico do agrupamento de BD onde foram realizados os nossos experimentos não era suficiente para o armazenamento de todos os objetos da nossa base de dados. Esse agrupamento de computadores faz parte de uma Grade localizada na França, específico para a realização de experimentos acadêmicos.

Mantendo o modelo lógico de dados, a base de dados da Amostra do CD2000 no ParGRES é formada por três tabelas de fatos (`CD00AMDOMI`, `CD00AMPESS` e `CD00AMFAMI`) e cada tabela possui 65, 107 e 28 atributos, respectivamente; e por 84 tabelas de dimensões (sendo 1 dimensão temporal e 10 dimensões espaciais), com 3 atributos cada uma. Os relacionamentos entre as tabelas de fatos e dimensões se mantêm, inclusive o relacionamento de hierarquia entre as dimensões espaciais.

Em relação ao modelo físico de dados, foram mantidos os índices primários (que representam as chaves primárias), os índices de junção (que formam as chaves de

junção entre as tabelas de fatos), e as chaves estrangeiras. Os índices secundários (nos demais atributos) não foram criados por questões de limitações de espaço físico em disco.

O processamento de consultas utilizando paralelismo intra-consulta no ParGRES é feito através da AVP, conforme descrito na seção 2.5.1 do Capítulo 2, que tem como requisito a existência de índices de agrupamento. Portanto, na base de dados da Amostra do CD2000 no ParGRES, foram definidos índices de agrupamento sobre o atributo CONTROLE das tabelas de fatos CD00AMDOMI, CD00AMPESS e CD00AMFAMI, além deste atributo ter sido escolhido para ser o atributo de fragmentação virtual dessas tabelas.

Os modelos de dados, lógico e físico, foram os mesmos utilizados em ambos os experimentos (com base totalmente replicada e parcialmente replicada). Nos experimentos com base totalmente replicada não há projeto de fragmentação, pois a base de dados da Amostra do CD2000 foi replicada em todos os nós. Nos experimentos com base parcialmente replicada, definimos o projeto de fragmentação e distribuição da base de dados do CD2000 seguindo a metodologia proposta na seção 3.1.2 do capítulo 3. As tabelas foram particionadas fisicamente em relação ao número de nós utilizando o atributo CONTROLE, também escolhido como atributo de fragmentação virtual, presente em todas as tabelas de fatos.

Na Tabela 4 apresentamos a média do número de registros de cada fragmento da tabela de fatos CD00AMDOMI, por Unidade da Federação, em diferentes configurações de nós; e o desvio padrão da distribuição em relação à média. Em um cenário com 2 nós, temos 2 fragmentos da relação CD00AMDOMI, e esses fragmentos possuem 2.652.357 e 2.652.354, respectivamente. A diferença entre o número de registros total é de 3 registros. Observando o número médio de registros de cada fragmento segundo a Unidade da Federação, percebemos, pelo desvio padrão, que o número de registros por fragmento é bem aproximado. Na Tabela 5 e na Tabela 6 também apresentamos a média do número de registros de cada fragmento das tabelas de fatos CD00AMFAMI e CD00AMPESS, por Unidade da Federação, em diferentes configurações de nós; e o desvio padrão da distribuição em relação à média.

É importante lembrar que apenas as três tabelas de fatos foram fragmentadas e distribuídas; todas as tabelas de dimensão foram replicadas entre os nós. Portanto, a decomposição das relações CD00AMDOMI, CD00AMPESS e CD00AMFAMI em fragmentos horizontais é definida da seguinte forma:

$$CD00AMDOMI_n = \sigma_{((n-1)(((max(CONTROLE))/n)+1) \leq CONTROLE \leq (n((max(CONTROLE))/n))} \textbf{(CD00AMDOMI)}$$

$$CD00AMFAMI_n = \sigma_{((n-1)(((max(CONTROLE))/n)+1) \leq CONTROLE \leq (n((max(CONTROLE))/n))} \textbf{(CD00AMFAMI)}$$

$$CD00AMPESS_n = \sigma_{((n-1)(((max(CONTROLE))/n)+1) \leq CONTROLE \leq (n((max(CONTROLE))/n))} \textbf{(CD00AMPESS)}$$

A replicação dos fragmentos foi feita alocando uma réplica de uma cópia primária (fragmento) no nó seguinte à localização da cópia primária.

Tabela 4: Média do número de registros por fragmento e desvio padrão (CD00AMDOMI)

CODUFCENSO	Total de registros	Cenário com 2 nós			Cenário com 4 nós			Cenário com 8 nós			Cenário com 16 nós			Cenário com 32 nós			Cenário com 64 nós		
		N.º de registros por fragmento			N.º de registros por fragmento			N.º de registros por fragmento			N.º de registros por fragmento			N.º de registros por fragmento			N.º de registros por fragmento		
		Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%
11	43293	21646,5	34,6	50	10823,3	64,7	25	5411,6	36,9	12,5	2705,8	46,8	6,25	1352,9	30,4	3,125	676,5	24,4	1,563
12	16818	8409,0	182,4	50	4204,5	86,1	25	2102,3	57,3	12,5	1051,1	37,3	6,25	525,6	25,1	3,125	262,8	16,4	1,563
13	63970	31985,0	117,4	50	15992,5	208,6	25	7996,3	103,9	12,5	3998,1	76,9	6,25	1999,1	51,0	3,125	999,5	33,9	1,563
14	9857	4928,5	87,0	50	2464,3	60,9	25	1232,1	36,2	12,5	616,1	19,5	6,25	308,0	15,4	3,125	154,0	11,6	1,563
15	145992	72996,0	155,6	50	36498,0	174,2	25	18249,0	160,9	12,5	9124,5	102,4	6,25	4562,3	66,4	3,125	2281,1	49,3	1,563
16	11821	5910,5	4,9	50	2955,3	24,0	25	1477,6	17,4	12,5	738,8	21,9	6,25	369,4	17,1	3,125	184,7	14,3	1,563
17	43043	21521,5	31,8	50	10760,8	90,5	25	5380,4	81,9	12,5	2690,2	59,7	6,25	1345,1	37,0	3,125	672,5	25,9	1,563
21	150441	75220,5	55,9	50	37610,3	109,5	25	18805,1	100,5	12,5	9402,6	107,1	6,25	4701,3	75,0	3,125	2350,6	46,8	1,563
22	94534	47267,0	157,0	50	23633,5	127,0	25	11816,8	100,0	12,5	5908,4	74,4	6,25	2954,2	49,7	3,125	1477,1	40,1	1,563
23	201143	100571,5	133,6	50	50285,8	136,3	25	25142,9	135,7	12,5	12571,4	110,2	6,25	6285,7	67,6	3,125	3142,9	60,5	1,563
24	92673	46336,5	111,0	50	23168,3	65,8	25	11584,1	80,8	12,5	5792,1	58,0	6,25	2896,0	35,6	3,125	1448,0	29,2	1,563
25	117577	58788,5	290,6	50	29394,3	156,6	25	14697,1	100,2	12,5	7348,6	80,6	6,25	3674,3	62,1	3,125	1837,1	45,6	1,563
26	225649	112824,5	166,2	50	56412,3	386,4	25	28206,1	216,0	12,5	14103,1	122,0	6,25	7051,5	74,4	3,125	3525,8	56,9	1,563
27	77896	38948,0	158,4	50	19474,0	119,2	25	9737,0	93,1	12,5	4868,5	71,9	6,25	2434,3	54,8	3,125	1217,1	31,6	1,563
28	55161	27580,5	293,4	50	13790,3	150,8	25	6895,1	111,0	12,5	3447,6	71,7	6,25	1723,8	39,7	3,125	861,9	32,3	1,563
29	378907	189453,5	283,5	50	94726,8	159,5	25	47363,4	198,3	12,5	23681,7	133,0	6,25	11840,8	102,2	3,125	5920,4	71,0	1,563
31	615101	307550,5	461,7	50	153775,3	301,5	25	76887,6	155,3	12,5	38443,8	154,8	6,25	19221,9	133,0	3,125	9611,0	94,8	1,563
32	98820	49410,0	292,7	50	24705,0	161,9	25	12352,5	83,4	12,5	6176,3	53,4	6,25	3088,1	46,1	3,125	1544,1	40,2	1,563
33	442976	221488,0	77,8	50	110744,0	86,3	25	55372,0	125,9	12,5	27686,0	149,9	6,25	13843,0	98,6	3,125	6921,5	73,8	1,563
35	1137154	568577,0	660,4	50	284288,5	378,4	25	142144,3	222,8	12,5	71072,1	151,3	6,25	35536,1	104,5	3,125	17768,0	89,4	1,563
41	336151	168075,5	388,2	50	84037,8	209,1	25	42018,9	228,6	12,5	21009,4	183,6	6,25	10504,7	120,0	3,125	5252,4	81,0	1,563
42	193633	96816,5	259,5	50	48408,3	216,1	25	24204,1	117,1	12,5	12102,1	83,2	6,25	6051,0	63,7	3,125	3025,5	47,0	1,563
43	365827	182913,5	200,1	50	91456,8	126,8	25	45728,4	161,7	12,5	22864,2	159,8	6,25	11432,1	117,3	3,125	5716,0	76,3	1,563
50	69401	34700,5	137,9	50	17350,3	57,4	25	8675,1	127,5	12,5	4337,6	68,1	6,25	2168,8	44,8	3,125	1084,4	31,1	1,563
51	86946	43473,0	446,9	50	21736,5	197,0	25	10868,3	115,3	12,5	5434,1	68,9	6,25	2717,1	51,6	3,125	1358,5	35,1	1,563
52	175132	87566,0	553,0	50	43783,0	227,3	25	21891,5	156,9	12,5	10945,8	103,4	6,25	5472,9	73,2	3,125	2736,4	47,3	1,563
53	54795	27397,5	13,4	50	13698,8	150,2	25	6849,4	76,4	12,5	3424,7	57,7	6,25	1712,3	45,6	3,125	856,2	31,2	1,563
Total por fragmento	**5.304.711**	**2.652.355,50**			**1.326.178,60**			**663.089,00**			**331.544,70**			**165.772,30**			**82.886,00**		

Tabela 5: Média do número de registros por fragmento e desvio padrão (CD00AMFAMI)

CODUFCENSO	Total de registros	Cenário com 2 nós			Cenário com 4 nós			Cenário com 8 nós			Cenário com 16 nós			Cenário com 32 nós			Cenário com 64 nós		
		Nº de registros por fragmento			Nº de registros por fragmento			Nº de registros por fragmento			Nº de registros por fragmento			Nº de registros por fragmento			Nº de registros por fragmento		
		Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%
11	45.856	22928,0	77,8	50	11464,0	60,0	25	5732,0	36,0	12,5	2866,0	47,5	6,25	1433,0	31,4	3,125	716,5	26,3	1,563
12	18.113	9056,5	178,9	50	4528,3	100,4	25	2264,1	63,8	12,5	1132,1	39,4	6,25	566,0	27,2	3,125	283,0	17,9	1,563
13	73.677	36838,5	176,1	50	18419,3	249,8	25	9209,6	127,2	12,5	4604,8	92,5	6,25	2302,4	59,4	3,125	1151,2	41,0	1,563
14	10.576	5288,0	99,0	50	2644,0	66,2	25	1322,0	36,8	12,5	661,0	20,0	6,25	330,5	17,1	3,125	165,3	12,2	1,563
15	166.831	83415,5	364,2	50	41707,8	268,8	25	20853,9	180,8	12,5	10426,9	118,1	6,25	5213,5	77,2	3,125	2606,7	55,2	1,563
16	13.338	6669,0	73,5	50	3334,5	36,6	25	1667,3	21,7	12,5	833,6	21,7	6,25	416,8	21,5	3,125	208,4	17,2	1,563
17	45.832	22916,0	79,2	50	11458,0	112,4	25	5729,0	86,6	12,5	2864,5	65,7	6,25	1432,3	40,8	3,125	716,1	27,9	1,563
21	166.793	83396,5	65,8	50	41698,3	152,6	25	20849,1	145,3	12,5	10424,6	132,8	6,25	5212,3	87,8	3,125	2606,1	56,6	1,563
22	103.284	51642,0	346,5	50	25821,0	167,4	25	12910,5	126,8	12,5	6455,3	94,0	6,25	3227,6	62,2	3,125	1613,8	47,9	1,563
23	219.843	109921,5	170,4	50	54960,8	199,5	25	27480,4	141,6	12,5	13740,2	128,6	6,25	6870,1	77,7	3,125	3435,0	68,3	1,563
24	102.556	51278,0	154,1	50	25639,0	104,7	25	12819,5	106,7	12,5	6409,8	71,6	6,25	3204,9	44,8	3,125	1602,4	35,4	1,563
25	128.391	64195,5	215,7	50	32097,8	177,5	25	16048,9	124,1	12,5	8024,4	90,8	6,25	4012,2	71,6	3,125	2006,1	53,8	1,563
26	247.081	123540,5	210,0	50	61770,3	488,0	25	30885,1	275,1	12,5	15442,6	153,6	6,25	7721,3	93,0	3,125	3860,6	67,5	1,563
27	85.836	42918,0	248,9	50	21459,0	143,5	25	10729,5	106,6	12,5	5364,8	90,9	6,25	2682,4	59,7	3,125	1341,2	35,3	1,563
28	60.248	30124,0	326,7	50	15062,0	149,5	25	7531,0	133,3	12,5	3765,5	83,6	6,25	1882,8	47,1	3,125	941,4	37,5	1,563
29	414.688	207344,0	164,0	50	103672,0	200,3	25	51836,0	256,8	12,5	25918,0	183,3	6,25	12959,0	124,8	3,125	6479,5	81,2	1,563
31	656.095	328047,5	620,1	50	164023,8	346,1	25	82011,9	170,3	12,5	41005,9	161,7	6,25	20503,0	140,3	3,125	10251,5	101,0	1,563
32	105.145	52572,5	379,7	50	26286,3	188,8	25	13143,1	92,9	12,5	6571,6	59,3	6,25	3285,8	53,2	3,125	1642,9	44,4	1,563
33	472.432	236216,0	142,8	50	118108,0	113,9	25	59054,0	185,3	12,5	29527,0	183,2	6,25	14763,5	118,9	3,125	7381,8	85,5	1,563
35	1.200.093	600046,5	468,8	50	300023,3	375,2	25	150011,6	224,9	12,5	75005,8	151,9	6,25	37502,9	107,9	3,125	18751,5	96,3	1,563
41	355.118	177559,0	247,5	50	88779,5	191,2	25	44389,8	223,7	12,5	22194,9	189,4	6,25	11097,4	130,3	3,125	5548,7	86,9	1,563
42	204.506	102253,0	123,0	50	51126,5	236,7	25	25563,3	117,4	12,5	12781,6	75,8	6,25	6390,8	70,3	3,125	3195,4	51,6	1,563
43	386.340	193170,0	427,1	50	96585,0	181,4	25	48292,5	156,6	12,5	24146,3	166,7	6,25	12073,1	127,8	3,125	6036,6	86,4	1,563
50	74.056	37028,0	101,8	50	18514,0	50,8	25	9257,0	110,3	12,5	4628,5	62,7	6,25	2314,3	43,1	3,125	1157,1	33,4	1,563
51	91.846	45923,0	369,1	50	22961,5	159,8	25	11480,8	106,2	12,5	5740,4	67,0	6,25	2870,2	50,3	3,125	1435,1	36,3	1,563
52	184.052	92026,0	591,1	50	46013,0	245,6	25	23006,5	163,8	12,5	11503,3	104,7	6,25	5751,6	81,3	3,125	2875,8	50,9	1,563
53	58.668	29334,0	59,4	50	14667,0	175,8	25	7333,5	96,0	12,5	3666,8	69,4	6,25	1833,4	51,1	3,125	916,7	35,1	1,563
Total por fragmento	**5.691.294,00**	**2.845.647,00**			**1.422.824,00**			**711.411,90**			**355.706,20**			**177.853,10**			**88.926,40**		

Tabela 6: Média do número de registros por fragmento e desvio padrão (CD00AMPESS)

		Cenário com 2 nós			Cenário com 4 nós			Cenário com 8 nós			Cenário com 16 nós			Cenário com 32 nós			Cenário com 64 nós		
CODUFCENSO	Total de registros	N.° de registros por fragmento			N.° de registros por fragmento			N.° de registros por fragmento			N.° de registros por fragmento			N.° de registros por fragmento			N.° de registros por fragmento		
		Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%	Média	Desvio Padrão	%
11	172.073	86036,5	74,2	50	43018,3	53,7	25	21509,1	98,1	12,5	10754,6	167,3	6,25	5377,3	116,2	3,125	2688,6	104,3	1,56
12	71.063	35531,5	712,1	50	17765,8	350,6	25	8882,9	256,7	12,5	4441,4	166,5	6,25	2220,7	115,1	3,125	1110,4	73,4	1,56
13	314.758	157379,0	263,0	50	78689,5	1498,5	25	39344,8	732,2	12,5	19672,4	437,5	6,25	9836,2	271,8	3,125	4918,1	184,0	1,56
14	41.639	20819,5	282,1	50	10409,8	260,7	25	5204,9	148,7	12,5	2602,4	97,9	6,25	1301,2	79,7	3,125	650,6	63,1	1,56
15	691.394	345697,0	789,1	50	172848,5	533,4	25	86424,3	618,3	12,5	43212,1	515,6	6,25	21606,1	369,0	3,125	10803,0	256,3	1,56
16	55.391	27695,5	188,8	50	13847,8	99,4	25	6923,9	91,8	12,5	3461,9	105,4	6,25	1731,0	83,7	3,125	865,5	69,1	1,56
17	175.904	87952,0	176,8	50	43976,0	453,9	25	21988,0	320,4	12,5	10994,0	246,7	6,25	5497,0	163,3	3,125	2748,5	110,7	1,56
21	703.621	351810,0	166,9	50	175905,0	514,1	25	87952,5	543,5	12,5	43976,3	566,8	6,25	21988,1	403,1	3,125	10994,1	260,7	1,56
22	405.936	202968,0	1237,4	50	101484,0	774,2	25	50742,0	501,4	12,5	25371,0	393,8	6,25	12685,5	274,7	3,125	6342,8	210,0	1,56
23	866.347	433173,5	1232,5	50	216586,8	692,0	25	108293,4	597,0	12,5	54146,7	538,0	6,25	27073,3	342,5	3,125	13536,7	290,0	1,56
24	390.126	195063,0	1410,0	50	97531,5	714,5	25	48765,8	452,8	12,5	24382,9	266,1	6,25	12191,4	171,9	3,125	6095,7	146,9	1,56
25	487.848	243924,0	1373,2	50	121962,0	789,9	25	60981,0	496,7	12,5	30490,5	377,4	6,25	15245,3	300,4	3,125	7622,6	213,1	1,56
26	935.536	467768,0	1100,3	50	233884,0	1795,0	25	116942,0	999,0	12,5	58471,0	537,8	6,25	29235,5	343,7	3,125	14617,8	258,7	1,56
27	348.429	174214,5	1482,8	50	87107,3	747,3	25	43553,6	401,1	12,5	21776,8	385,6	6,25	10888,4	280,9	3,125	5444,2	164,9	1,56
28	230.984	115492,0	1081,9	50	57746,0	540,8	25	28873,0	431,1	12,5	14436,5	300,5	6,25	7218,3	174,6	3,125	3609,1	149,9	1,56
29	1.598.126	799063,0	775,0	50	399531,5	566,9	25	199765,8	848,0	12,5	99882,9	685,0	6,25	49941,4	519,8	3,125	24970,7	362,6	1,56
31	2.347.758	1173879,0	2819,9	50	586939,5	1466,1	25	293469,8	732,9	12,5	146734,9	594,8	6,25	73367,4	539,4	3,125	36683,7	399,6	1,56
32	369.666	184833,0	1479,3	50	92416,5	745,2	25	46208,3	412,8	12,5	23104,1	258,1	6,25	11552,1	211,7	3,125	5776,0	163,4	1,56
33	1.511.640	755820,0	1438,3	50	377910,0	716,9	25	188955,0	615,6	12,5	94477,5	524,3	6,25	47238,8	370,0	3,125	23619,4	297,2	1,56
35	4.038.217	2019108,5	594,7	50	1009554,3	1168,0	25	504777,1	685,7	12,5	252388,6	537,2	6,25	126194,3	436,3	3,125	63097,1	380,7	1,56
41	1.218.361	609180,5	1536,5	50	304590,3	873,8	25	152295,1	736,9	12,5	76147,6	573,2	6,25	38073,8	400,4	3,125	19036,9	284,5	1,56
42	693.703	346851,5	40,3	50	173425,8	1028,8	25	86712,9	519,7	12,5	43356,4	340,8	6,25	21678,2	251,6	3,125	10839,1	195,5	1,56
43	1.209.631	604815,5	997,7	50	302407,8	879,2	25	151203,9	738,0	12,5	75601,9	626,1	6,25	37801,0	415,4	3,125	18900,5	286,3	1,56
50	251.403	125701,5	275,1	50	62850,8	162,2	25	31425,4	443,6	12,5	15712,7	234,3	6,25	7856,3	165,3	3,125	3928,2	129,2	1,56
51	326.022	163011,0	1486,3	50	81505,5	618,4	25	40752,8	353,4	12,5	20376,4	285,9	6,25	10188,2	210,5	3,125	5094,1	146,1	1,56
52	617.948	308974,0	1521,7	50	154487,0	738,7	25	77243,5	516,8	12,5	38621,8	320,4	6,25	19310,9	283,2	3,125	9655,4	180,8	1,56
53	200.888	100444,0	308,3	50	50222,0	650,7	25	25111,0	348,8	12,5	12555,5	236,2	6,25	6277,8	167,6	3,125	3138,9	125,0	1,56
Total por fragmento	**20.274.412,00**	**10.137.205,50**			**5.068.603,30**			**2.534.301,80**			**1.267.150,80**			**633.575,50**			**316.787,70**		

5 Análise de desempenho

Neste capítulo são descritos os experimentos realizados para a análise de desempenho de uma base de dados do BME gerenciada pelo ParGRES em um agrupamento de computadores. Na seção 5.1 apresentamos o ambiente computacional de execução dos experimentos, as especificações da base de dados e as consultas utilizadas. Na seção 5.2 descrevemos como os experimentos foram realizados e os resultados obtidos.

5.1 Ambiente de execução

5.1.1 Ambiente computacional

Todos os experimentos foram executados em um agrupamento de banco de dados de 64 nós do projeto Grid5000 [11] situado em Rennes, na França. Este projeto tem como objetivo disponibilizar uma infra-estrutura de larga escala (Grade) que pode ser reconfigurada, controlada e monitorada para a realização de experimentos acadêmicos de processamento paralelo e distribuído. Cada nó do agrupamento utilizado tem dois processadores Intel Xeon com 2.3GHz, 4GB de memória RAM, um disco rígido de 160GB e uma placa de rede Myri-10G. Utilizamos o SGBD PostgreSQL 8.2.4 [30] para gerenciar a base de dados do experimento e o sistema operacional Debian Linux versão SID para amd64. A Tabela 7 apresenta características detalhadas do agrupamento de computadores PARAQUAD, onde foram realizados todos os experimentos.

Tabela 7: Características detalhadas do agrupamento de computadores PARAQUAD

Agrupamento de Computadores PARAQUAD	
Modelo	Dell Power Edge 1950 64 nós
CPU	Intel Xeon 5148 LV - 2.33 Ghz - 64 nós x 2 cpus por nó = 128 cpus - 132 cpus x 2 cores por cpu = 264 nós
Memória	4 GB/nó
Rede	64 cartões Myri-10G (10G - PCIE - 8AC) Gigabit Ethernet Driver bn x2
Disco Rígido	160 GB / SATA

5.1.2 Base de Dados

Nesta dissertação, nomeamos a base de dados do Censo Demográfico 2000 como AmCD2000, e a mesma foi gerada de acordo com as especificações do BME. Nesta base de dados foram definidas três tabelas de fatos, seguindo a organização de temas da pesquisa: CD00AMDOMI (dados de Domicílios), CD00AMPESS (dados de Pessoas) e CD00AMFAMI (dados de Família). Essas tabelas possuem 5.304.711, 20.274.412 e 5.691.294 registros, respectivamente. As tabelas de fato tiveram suas tuplas ordenadas fisicamente segundo o atributo de fragmentação virtual (AFV) de cada uma e um índice foi criado sobre cada um destes atributos. O atributo CONTROLE, que compõe a chave primária das tabelas de fato, foi definido como AFV. É importante salientar que esse atributo não apresenta distorção de dados, ou seja, a distribuição de valores é sequencial e uniforme. As tabelas de fatos relacionam-se entre si através de uma chave de junção composta pelas suas chaves primárias.

Também foram definidas as 84 tabelas de dimensões, incluindo a dimensão temporal (T004) e as dez dimensões espaciais (G000, G031, G032, G033, G034, G035, G036, G039 e G042), seguindo a organização dessas tabelas apresentada na seção 4.2.2. A chave primária de cada tabela de dimensão é o atributo CODIGO (presente em todas essas tabelas). As tabelas de dimensões se relacionam com as tabelas de fatos através de chaves estrangeiras.

A seguir, apresentamos as 24 dimensões que se relacionam com CD00AMDOMI e suas cardinalidades: |M003| = 12, |M075| = 3, |M078| = 5, |M102| = 4, |M103| = 7, |M104| = 4, |M105| = 4, |M106| = 4, |M109| = 8, |M115| = 8, |M116| = 8, |M128| = 6, |M129| = 8, |M208| = 2, |M209| = 12, |M233| = 5, |M270| = 3, |M272| = 10, |M273| = 11, |M274| = 7, |M275| = 11, |M276| = 11, |M277| = 11 e |M278| = 11.

As 7 dimensões que se relacionam com CD00AMFAMI e suas cardinalidades são: |M290| = 16, |M291| = 17, |M292| = 17, |M293| = 8, |M295| = 13, |M296| = 16 e |M297| = 13.

As 42 dimensões que se relacionam com CD00AMPESS e suas cardinalidades são: |M159| = 7, |M167| = 4, |M298| = 13, |M300| = 2, |M301| = 13, |M302| = 21, |M306| = 144, |M307| = 54, |M308| = 6, |M309| = 7, |M311| = 3, |M314| = 8, |M315| = 5510, |M320| = 3, |M321| = 5, |M322| = 15, |M323| = 11, |M324| = 12, |M325| = 13, |M326| = 5, |M327| = 63, |M330| = 5, |M331| = 7, |M332| = 7, |M333| = 4, |M334| = 513, |M338| = 224, |M341| = 10, |M342| = 7, |M343| = 5, |M348| = 15, |M355| = 3, |M361| = 8, |M4210| = 98, |M4219| =

260, |M4230| = 99, |M4239| = 261, |M4276| = 5605, |M4279| = 232, |M4300| = 21, |M4354| = 94 e |M4511| = 3.

O tamanho da base de dados AmCD2000 é de aproximadamente 20 GB, sem contar o espaço ocupado por índices definidos sobre os atributos. Por questões de limitação de espaço físico no agrupamento de computadores (onde foram realizados os experimentos), alguns índices não foram definidos. Assim, criamos índices apenas sobre o AFV, requisito essencial para a AVP. Finalmente, atualizamos as estatísticas do banco de dados para serem utilizadas pelo otimizador do SGBD.

Nos Anexos A, B e C apresentamos informações referentes aos atributos que compõem as tabelas de fatos; às descrições de todas as dimensões e seus atributos; e às dimensões e atributos referenciadas nas chaves estrangeiras das tabelas de fatos.

5.1.3 Consultas

Para a realização dos experimentos de análise de desempenho do ParGRES, foram selecionadas 14 consultas de alto custo tipicamente realizadas sobre a base de dados do CD2000 no BME, com diferentes níveis de complexidade. As 14 consultas foram apresentadas na seção 4.2.4.

As consultas do BME são executadas em dois passos conforme descrito na seção 4.2.4. Para os experimentos, as consultas foram modificadas para serem realizadas em um único passo. Essas consultas foram reescritas mesclando o passo 1 com o passo 2. As funções específicas do SGBD paralelo comercial foram excluídas. Seja uma consulta C gerada pelo BME:

PASSO 1:

```
select 76 geografia_cod, 9000 geografia_dim_cod, cd00amdomi.codanopesq
tempo_cod, 8005 tempo_dim_cod, nvl(cd00amdomi.codv1006, -1) col1,
sum(cd00amdomi.peso) frequencia, count(*) contagem
from ibge.cd00amdomi
group by 76, cd00amdomi.codanopesq, nvl(cd00amdomi.codv1006, -1)
```

PASSO 2:

```
select bmm_tmp_211070.geografia_cod, bmm_tmp_211070.geografia_dim_cod,
g000.denominacao geografia_desc, bmm_tmp_211070.tempo_cod,
bmm_tmp_211070.tempo_dim_cod, t005.denominacao tempo_desc, tab1.denominacao
col1, tab1.ordem col1_ord, tab1.codigo col1_cod, bmm_tmp_211070.frequencia,
bmm_tmp_211070.contagem
from g000, t004, m208 tab1, bmm_tmp_211070
where bmm_tmp_211070.geografia_dim_cod = 9000 and (bmm_tmp_211070.tempo_cod
= '2000') and bmm_tmp_211070.tempo_dim_cod = 8004 and bmm_tmp_211070.col1 =
tab1.codigo and bmm_tmp_211070.tempo_cod = t004.codigo
```

A consulta C foi reescrita da seguinte maneira para a realização dos experimentos:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8004 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00amdomi.codv1006 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00amdomi.pesodomi) as frequencia,
   count(*) as contagem
from g000, t004, m208 as tab1, cd00amdomi
where cd00amdomi.codanopesq = 2000
   and cd00amdomi.codv1006 = tab1.codigo
   and cd00amdomi.codanopesq = t004.codigo
group by g000.denominacao, cd00amdomi.codanopesq, t004.denominacao,
   cd00amdomi.codv1006, tab1.denominacao, tab1.ordem, tab1.codigo;
```

A seguir, apresentamos as versões em um só passo das consultas `Q1` a `Q14`:

Q1:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00amdomi.codv1006 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00amdomi.pesodomi) as frequencia,
   count(*) as contagem
from g000, t004, m208 as tab1, cd00amdomi
where cd00amdomi.codanopesq = 2000
   and cd00amdomi.codv1006 = tab1.codigo
   and cd00amdomi.codanopesq = t004.codigo
group by g000.denominacao, cd00amdomi.codanopesq, t004.denominacao,
cd00amdomi.codv1006, tab1.denominacao, tab1.ordem, tab1.codigo;
```

Q2:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv4752 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g000, t004, m302 as tab1, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codv4752 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
group by g000.denominacao, cd00ampess.codanopesq ,t004.denominacao,
cd00ampess.codv4752,tab1.denominacao,tab1.ordem,tab1.codigo;
```

Q3:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   sum(cd00ampess.v4752 * cd00ampess.pesopess) / sum(cd00ampess.pesopess) as
col1,
   sum(cd00ampess.pesopess)as frequencia,
   count(*) as contagem
from g000, t004, cd00ampess
where cd00ampess.codanopesq = 2000
```

```
   and cd00ampess.codanopesq = t004.codigo
group by g000.denominacao, cd00ampess.codanopesq, t004.denominacao;
```

Q4:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv4690 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.v4690 * cd00ampess.pesopess) as col2,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g000, t004, m348 as tab1, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codv4690 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
group by g000.denominacao, cd00ampess.codanopesq, t004.denominacao,
cd00ampess.codv4690, tab1.codigo, tab1.denominacao, tab1.ordem;
```

Q5:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv4090 as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g000, t004, m307 as tab1, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codv4090 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
   and (cd00ampess.codv4090 = 0 or cd00ampess.codv4090 = 110 or
cd00ampess.codv4090 = 240 or cd00ampess.codv4090 = 590 or
cd00ampess.codv4090 = 750 or cd00ampess.codv4090 = 810 or
cd00ampess.codv4090 = 850)
group by g000.denominacao, cd00ampess.codanopesq, t004.denominacao,
cd00ampess.codv4090, tab1.ordem, tab1.codigo;
```

Q6:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   sum(cd00ampess.v4752 * cd00ampess.pesopess) / sum(cd00ampess.pesopess) as
col1,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g000, t004, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codanopesq = t004.codigo
   and (cd00ampess.v4752 between 20 and 40)
group by g000.denominacao, cd00ampess.codanopesq, t004.denominacao;
```

Q7:

```
select cd00ampess.codmunicipio as geografia_cod,
   9035 as geografia_dim_cod,
   g035.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0401 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
```

```
   sum (cd00ampess.codv4300 * cd00ampess.pesopess) /
sum(cd00ampess.pesopess) as col2,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g035, t004, m300 as tab1, cd00ampess
where cd00ampess.codmunicipio = 3304557
   and cd00ampess.codanopesq = 2000
   and cd00ampess.codv0401 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
   and cd00ampess.codmunicipio = g035.codigo
   and g035.ind_exibicao = 's'
group by cd00ampess.codmunicipio , g035.denominacao, cd00ampess.codanopesq,
t004.denominacao, cd00ampess.codv0401, tab1.denominacao, tab1.ordem,
tab1.codigo;
```

Q8:

```
select cd00ampess.codmunicipio as geografia_cod,
   9035 as geografia_dim_cod,
   g035.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0401 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.codv4300 * cd00ampess.pesopess) / sum(cd00ampess.pesopess)
as col2,
   sum(cd00ampess.pesopess) as frequencia, count(*) as contagem
from g035, t004, m300 as tab1, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codv0401 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
   and cd00ampess.codmunicipio = g035.codigo
   and (cd00ampess.codmunicipio = 2408102 or cd00ampess.codmunicipio =
3304557)
   and g035.ind_exibicao = 's'
group by cd00ampess.codmunicipio, g035.denominacao, cd00ampess.codanopesq,
t004.denominacao, cd00ampess.codv0401, tab1.denominacao, tab1.ordem,
tab1.codigo;
```

Q9:

```
select 76 as geografia_cod,
   9000 as geografia_dim_cod,
   g000.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0401 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   cd00amdomi.codv1006 as col2,
   tab2.denominacao as col2,
   tab2.ordem as col2_ord,
   tab2.codigo as col2_cod,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g000, t004, m300 as tab1, m208 as tab2, cd00ampess, cd00amdomi
where (cd00ampess.controle = cd00amdomi.controle)
   and cd00amdomi.codanopesq = 2000
   and cd00ampess.codv0401 = tab1.codigo
   and cd00amdomi.codv1006 = tab2.codigo
   and cd00amdomi.codanopesq = t004.codigo
group by g000.denominacao, cd00amdomi.codanopesq, t004.denominacao,
cd00ampess.codv0401, tab1.denominacao, tab1.ordem, tab1.codigo,
cd00amdomi.codv1006, tab2.denominacao, tab2.ordem, tab2.codigo;
```

Q10:

```
select cd00amdomi.codmunicipio as geografia_cod,
   9035 as geografia_dim_cod,
   g035.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0401 as col1,
   tab1.denominacao as col1,
```

```
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   cd00amdomi.codv1006 as col2,
   tab2.denominacao as col2,
   tab2.ordem as col2_ord,
   tab2.codigo as col2_cod,
   sum(cd00ampess.codv4300 * cd00ampess.pesopess) / sum(cd00ampess.pesopess)
as col3,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g035, t004, m300 as tab1, m208 as tab2, cd00ampess, cd00amdomi
where (cd00ampess.codmunicipio = 3304557)
   and g035.ind_exibicao = 's'
   and cd00ampess.controle = cd00amdomi.controle
   and cd00amdomi.codanopesq = 2000
   and cd00ampess.codv0401 = tab1.codigo
   and cd00amdomi.codv1006 = tab2.codigo
   and cd00amdomi.codanopesq = t004.codigo
   and cd00amdomi.codmunicipio = g035.codigo
group by cd00amdomi.codmunicipio, g035.denominacao, cd00amdomi.codanopesq,
t004.denominacao, cd00ampess.codv0401, tab1.denominacao,
cd00amdomi.codv1006, tab2.denominacao, tab1.ordem, tab1.codigo, tab2.ordem,
tab2.codigo;
```

Q11:

```
select cd00amdomi.codmunicipio as geografia_cod,
   9035 as geografia_dim_cod,
   g035.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0401 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   cd00amdomi.codv1006 as col2,
   tab2.denominacao as col2,
   tab2.ordem as col2_ord,
   tab2.codigo as col2_cod,
   sum(cd00ampess.codv4300 * cd00ampess.pesopess) / sum(cd00ampess.pesopess)
as col3,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g035, t004, m300 as tab1, m208 as tab2, cd00ampess, cd00amdomi
where (cd00ampess.codmunicipio = 2408102 or cd00ampess.codmunicipio =
3304557)
    and g035.ind_exibicao = 's'
    and cd00ampess.controle = cd00amdomi.controle
   and cd00amdomi.codanopesq = 2000
   and cd00ampess.codv0401 = tab1.codigo
   and cd00amdomi.codv1006 = tab2.codigo
   and cd00amdomi.codanopesq = t004.codigo
   and cd00amdomi.codmunicipio = g035.codigo
group by cd00amdomi.codmunicipio, g035.denominacao, cd00amdomi.codanopesq,
t004.denominacao, cd00ampess.codv0401, tab1.denominacao, tab1.ordem,
tab1.codigo, cd00amdomi.codv1006, tab2.denominacao, tab2.ordem, tab2.codigo;
```

Q12:

```
select cd00amdomi.codufcenso as geografia_cod,
   9032 as geografia_dim_cod,
   g032.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00amdomi.codv0201 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.v4752 * cd00ampess.pesopess) / sum(cd00ampess.pesopess) as
col2,
   sum(cd00amfami.v4616_7400 * cd00amfami.pesofami) /
sum(cd00amfami.pesofami) as col3,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g032, t004, m270 as tab1, cd00amfami, cd00ampess, cd00amdomi
where cd00amdomi.codv0201 = 1
```

```
   and cd00ampess.v4752 between 7 and 14
   and cd00amfami.v4616_7400 <= 121
   and (cd00ampess.controle = cd00amfami.controle and cd00ampess.v0404 =
cd00amfami.v0404 and cd00amfami.controle = cd00amdomi.controle)
   and cd00amdomi.codanopesq = 2000
   and cd00amdomi.codv0201 = tab1.codigo
   and cd00amdomi.codanopesq = t004.codigo
   and cd00amdomi.codufcenso = g032.codigo
   and g032.ind_exibicao = 's'
group by cd00amdomi.codufcenso, g032.denominacao, cd00amdomi.codanopesq,
t004.denominacao, cd00amdomi.codv0201, tab1.codigo, tab1.denominacao,
tab1.ordem;
```

Q13:

```
select cd00ampess.codufcenso as geografia_cod,
   9032 as geografia_dim_cod,
   g032.denominacao as geografia_desc,
   cd00ampess.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00ampess.codv0428 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g032, t004, m320 as tab1, cd00ampess
where cd00ampess.codanopesq = 2000
   and cd00ampess.codv0428 = tab1.codigo
   and cd00ampess.codanopesq = t004.codigo
   and cd00ampess.codufcenso = g032.codigo
   and g032.ind_exibicao = 's'
group by cd00ampess.codufcenso, g032.denominacao, cd00ampess.codanopesq,
t004.denominacao, cd00ampess.codv0428, tab1.codigo, tab1.denominacao,
tab1.ordem;
```

Q14:

```
select cd00amdomi.codufcenso as geografia_cod,
   9032 as geografia_dim_cod,
   g032.denominacao as geografia_desc,
   cd00amdomi.codanopesq as tempo_cod,
   8005 as tempo_dim_cod,
   t004.denominacao as tempo_desc,
   cd00amfami.codv0404_2 as col1,
   tab1.denominacao as col1,
   tab1.ordem as col1_ord,
   tab1.codigo as col1_cod,
   cd00amdomi.codv1006 as col2,
   tab2.denominacao as col2,
   tab2.ordem as col2_ord,
   tab2.codigo as col2_cod,
   cd00ampess.codv0412 as col3,
   tab3.denominacao as col3,
   tab3.ordem as col3_ord,
   tab3.codigo as col3_cod,
   cd00ampess.codv0410 as col4,
   tab4.denominacao as col4,
   tab4.ordem as col4_ord,
   tab4.codigo as col4_cod,
   sum(cd00ampess.pesopess) as frequencia,
   count(*) as contagem
from g032, t004, m295 as tab1, m208 as tab2, m308 as tab3, m361 as tab4,
cd00amfami, cd00ampess, cd00amdomi
where g032.ind_exibicao = 's'
   and (cd00ampess.codv0412 = 1 or cd00ampess.codv0410 = 1)
   and (cd00ampess.controle = cd00amfami.controle and cd00ampess.v0404 =
cd00amfami.v0404 and cd00amfami.controle = cd00amdomi.controle)
   and cd00amdomi.codanopesq = 2000
   and cd00amfami.codv0404_2 = tab1.codigo
   and cd00amdomi.codv1006 = tab2.codigo
   and cd00ampess.codv0412 = tab3.codigo
   and cd00ampess.codv0410 = tab4.codigo
   and cd00amdomi.codanopesq = t004.codigo
   and cd00amdomi.codufcenso = g032.codigo
group by cd00amdomi.codufcenso, g032.denominacao, cd00amdomi.codanopesq,
t004.denominacao, cd00amfami.codv0404_2, tab1.denominacao, tab1.ordem,
```

```
tab1.codigo, cd00amdomi.codv1006, tab2.denominacao, tab2.ordem, tab2.codigo,
cd00ampess.codv0412, tab3.denominacao, tab3.ordem, tab3.codigo,
cd00ampess.codv0410, tab4.denominacao, tab4.ordem, tab4.codigo;
```

As principais características das consultas podem ser visualizadas na Tabela 8.

Tabela 8: Principais características das consultas.

Consulta	Tabela(s) Acessada(s)	Número de Agregações	Agrupamento	Predicado	Junção com Dimensão	Junção com Tabela de Fatos
Q1	CD00AMDOMI, G000,T004,M208	2	S	N	S	N
Q2	CD00AMPESS, G000,T004,G032	2	S	N	S	N
Q3	CD00AMPESS, G000,T004	4	S	N	S	N
Q4	CD00AMPESS, G000,M004,M348	3	S	N	S	N
Q5	CD00AMPESS, G000, T004, M307	2	S	S	S	N
Q6	CD00AMPESS, G000, T004	4	S	S	S	N
Q7	CD00AMPESS, G035, T004, M300	4	S	S	S	N
Q8	CD00AMPESS, G035, T004, M300	4	S	S	S	N
Q9	CD00AMPESS, CD00AMDOMI, G000, T004, M208, M300	2	S	N	S	S
Q10	CD00AMPESS, CD00AMDOMI, G035, T004, M300	4	S	S	S	S
Q11	CD00AMPESS, CD00AMDOMI, G035, T004, M300	4	S	S	S	S
Q12	CD00AMFAMI, CD00AMDOMI, CD00AMPESS, G032, T004, M270	4	S	S	S	S
Q13	CD00AMPESS, CD00AMDOMI, G032, T004, M208, M320	2	S	N	S	S
Q14	CD00AMDOMI, CD00AMPESS, CD00AMFAMI, G032, T004, M208, M295,M357, M079	2	S	S	S	S

5.2 Experimentos de aceleração

Nesta seção são apresentados os resultados obtidos nos experimentos de aceleração. O objetivo desses experimentos é avaliar o desempenho do processamento de consultas OLAP sobre uma base de dados reais, utilizando paralelismo inter e intra-consulta gerenciado pelo ParGRES. Para tanto, foram conduzidos dois tipos de experimentos: (i) com replicação total da base de dados nos nós do agrupamento de computadores; e (ii) com replicação parcial da base de dados nos nós do agrupamento de computadores. Esses experimentos foram divididos em duas etapas: na primeira, as consultas foram executadas isoladamente com diferentes números de nós do agrupamento, e, na segunda, as consultas foram organizadas em lotes e executadas concorrentemente, também com diferentes números de nós. O número de nós variou entre 1 e 64.

É importante lembrar que as consultas foram selecionadas levando-se em conta algumas características que aumentam ou diminuem a complexidade e o custo de sua execução. Dentre as características, podemos citar o número de registros lidos, número de agregações, número de junções e número de fragmentos acessados. Nos experimentos com replicação total da base, as consultas que apresentavam características relacionadas ao custo de acesso a fragmentos não tiveram seu desempenho afetado por essa variável, pois os dados estavam todos no mesmo computador e disco.

Para avaliar o desempenho do processamento paralelo de consultas, utilizamos como métrica de desempenho os tempos de execução das consultas e as variações entre estes tempos. O tempo de execução de uma consulta é o tempo transcorrido desde o envio da consulta para o SGBD até o recebimento do resultado pela aplicação cliente. Esse tempo é denominado "tempo de sala"[7], ou seja, começamos a contar o tempo no momento em que submetemos a consulta e paramos quando o resultado é retornado.

Alguns eventos aleatórios de sistema podem ocorrer durante o processamento paralelo de consultas, interferindo o tempo de processamento. Neste sentido, é recomendado que o tempo de processamento de uma mesma consulta seja medido mais de uma vez, a fim de possibilitar uma análise na variação de tempos obtidos, detectando se a causa dessa variação de tempos é um evento isolado ou um problema real que ocorre durante o processamento, no envio da consulta ou no recebimento do resultado.

[7] CROWL[8] denomina esse tempo como *Elapsed (wall clock) time*.

Para verificar o tamanho da variação de tempo, calculamos o desvio padrão dos tempos de execução em relação à média, com base em todos os argumentos fornecidos; e a proporção de valores caindo dentro de um, dois e três desvios padrões da média [8]. CROWL [8] recomenda a utilização do gráfico com barras de erro caso exista uma variação significativa nos dados com distribuição normal. Esses gráficos encontram-se no Anexo D.

A aceleração linear é uma medida de desempenho derivada do tempo de execução de uma consulta processada em relação ao número de processadores utilizados para executarem essa consulta. Existem quatro tipos de aceleração: aceleração linear normal, aceleração linear escalar, aceleração linear com limite de memória e aceleração linear de precisão. Para analisar e comparar nossos resultados, utilizamos a aceleração linear normal, que é calculada dividindo o tempo de execução sequencial (com um nó) pelo tempo de execução com um determinado número de nós. Utiliza-se a seguinte notação para representar a aceleração linear:

$$Al = T_1 / T_p$$

tal que:

Al = aceleração linear

T_1 = tempo de execução sequencial (processamento com um nó)

p = número de processadores para executar uma consulta

T_p = tempo de execução com p processadores

Se a divisão entre os tempos (**Al**) é igual a p, alcançamos aceleração linear; se **Al** é maior do que **p**, alcançamos aceleração super linear; e se **Al** é menor do que **p**, alcançamos aceleração sub-linear.

As métricas de desempenho utilizadas nesta dissertação foram seguidas de CROWL [8], que descreve um conjunto de regras para medir, apresentar, analisar e comparar o desempenho de processamento paralelo, a fim de evitar uma má interpretação dos resultados e facilitar a leitura de tabelas e gráficos que apresentam os dados gerados.

Para avaliar o comportamento do ParGRES durante o balanceamento de carga e o quanto a difusão de mensagens entre os nós afetam no tempo de execução, também coletamos o número de mensagens de oferta e de aceite de ajuda enviadas pelos nós. Por ser mais complexa, realizamos essa análise apenas para as consultas Q9 e Q12, pois suas características refletem as características das outras consultas.

Na seção 5.2.1 são apresentados os experimentos com replicação total e os seus resultados. Na seção 5.2.2 são apresentados os experimentos com replicação parcial e seus resultados. Os resultados obtidos em ambos os experimentos nos motivaram a realizar um experimento adicional, descrito na seção 5.2.3.

5.2.1 Experimentos com Replicação Total

Esses experimentos têm como finalidade avaliar o desempenho do processamento de consultas utilizando o ParGRES com replicação total da base de dados `AmCD2000` nos 64 nós que compõem o agrupamento de banco de dados. Em cada nó existe uma instância do SGBD PostgreSQL instalada para gerenciar cada cópia da base de dados.

5.2.1.1 Experimentos com consultas isoladas

Essa primeira série de experimentos foi realizada para avaliar o desempenho do processamento individual de consultas com as diferentes configurações de agrupamento de banco de dados. As 14 consultas foram executadas cinco vezes em sequência com 1, 2, 4, 8, 16, 32 e 64 nós e seus tempos coletados. Em seguida, foi calculada a média entre os tempos de cada consulta em cada configuração e apresentado como tempo final de processamento. Entre as cinco execuções de cada consulta, detectamos uma variação entre os tempos coletados. Seguindo a recomendação de CROWL[8] analisamos essa variação de tempo através do cálculo do desvio padrão dos tempos em relação à média. Essas informações nos permitiram concluir que todos os tempos coletados (100%) estão entre dois desvios padrões da média, nos levando a ter uma distribuição normal dos tempos de processamento de cada consulta por número de nós. Portanto, podemos utilizar a média entre os tempos de processamentos como tempo final, sem que esta venha a prejudicar nossa análise de desempenho proposta. A variação entre os tempos de execução de cada consulta por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada consulta por número de nós.

Resultados

Todas as consultas foram processadas utilizando paralelismo intra-consulta. A Tabela 9 apresenta os tempos médios de processamento de cada consulta em cada

configuração do agrupamento de banco de dados. Os tempos médios são apresentados em segundos.

Tabela 9: Tempo médio de execução das consultas (em segundos) por número de nós

	Número de nós						
Consulta	1	2	4	8	16	32	64
Q1	31,90	13,59	9,00	7,86	2,61	2,50	2,36
Q2	217,41	113,30	24,41	17,85	13,70	7,18	2,50
Q3	217,47	107,60	16,81	13,96	7,94	5,78	2,19
Q4	217,90	101,76	16,89	12,88	12,36	4,79	2,24
Q5	217,44	88,77	10,70	10,38	9,12	4,47	1,69
Q6	217,62	92,48	11,45	11,42	7,44	1,91	1,75
Q7	217,76	85,98	12,33	10,55	5,60	1,51	1,35
Q8	217,67	86,19	10,08	9,01	8,35	2,30	1,81
Q9	460,30	208,19	41,30	22,52	11,03	5,71	3,60
Q10	586,65	244,96	16,18	10,86	8,86	5,70	1,70
Q11	568,54	273,66	13,35	12,74	11,54	2,30	1,82
Q12	1.242,22	618,85	174,95	84,45	42,25	21,91	11,43
Q13	218,37	112,56	23,15	11,68	8,15	5,83	2,94
Q14	1.030,08	511,11	93,80	48,08	25,50	14,45	9,20

A Tabela 9 mostra que os resultados obtidos nesse experimento foram bons, porque todas as consultas apresentaram redução no tempo de processamento à medida que se aumentou o número de nós. Para melhor visualizá-los, a Figura 16, a Figura 17 e a Figura 18 apresentam um gráfico com os tempos de execução normalizados. Os tempos normalizados foram obtidos dividindo o tempo de execução de cada consulta pelo seu maior tempo de execução, no nosso caso, o tempo sequencial da consulta (tempo de processamento de uma consulta com um nó).

Com exceção da consulta `Q1`, que é uma consulta de rápido processamento, as consultas obtiveram aceleração super-linear em todas as configurações do agrupamento. Com configurações de 2 e 4 nós, ocorre uma redução significativa no tempo de processamento de todas as consultas, sendo as maiores acelerações super-lineares obtidas. O que explica essa aceleração é o fato de todos os fragmentos virtuais já poderem ser carregados na memória principal dos nós. Com configurações superiores, como 8, 16, 32 e 64 nós, a redução no tempo de processamento é menor, mas ainda assim obtêm-se aceleração super-linear em todos os casos. As consultas `Q12` e `Q14` são as consultas com maior tempo de processamento, 21 e 18 minutos, respectivamente. Com 8 nós, o tempo de `Q12` diminui para 1,41 minutos e o tempo de `Q14` para 48 segundos. E com 64 nós, os tempos de `Q12` e `Q14` diminuem para apenas 11 e 9 segundos, respectivamente.

Na Figura 19 e na Figura 20 apresentamos as curvas de aceleração super-linear obtidas nesses experimentos. Esses gráficos foram construídos utilizando escala logarítmica nos eixos `x` e `y`. O eixo `x` representa o número de nós do agrupamento e o eixo `y`, a aceleração. Para facilitar a visualização da aceleração super-linear, foi plotada uma linha reta e grossa, representando a curva de aceleração linear. No gráfico da Figura 19 pode-se visualizar que apenas a consulta `Q1` não alcançou aceleração linear.

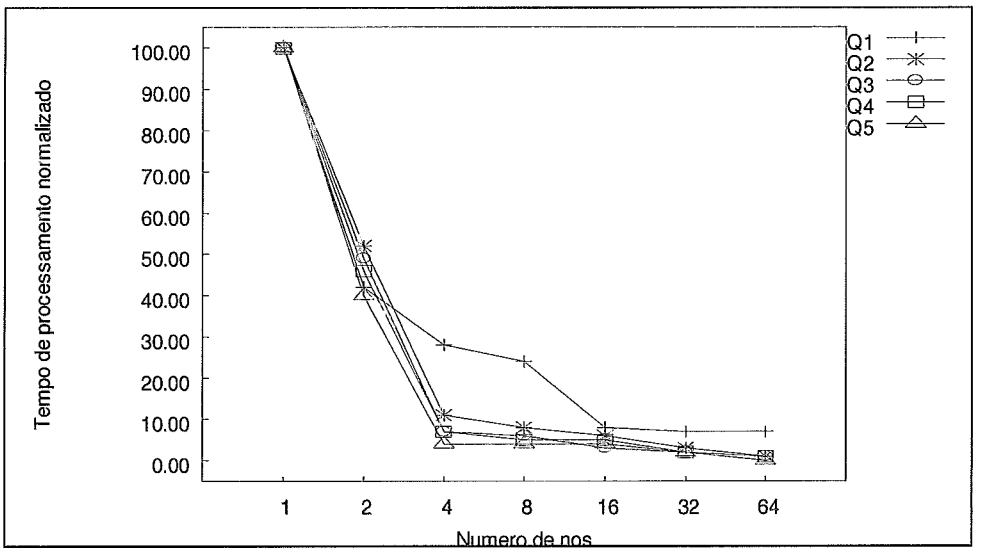


Figura 16: Tempos de execução normalizados por número de nós - Consultas Q1 a Q5

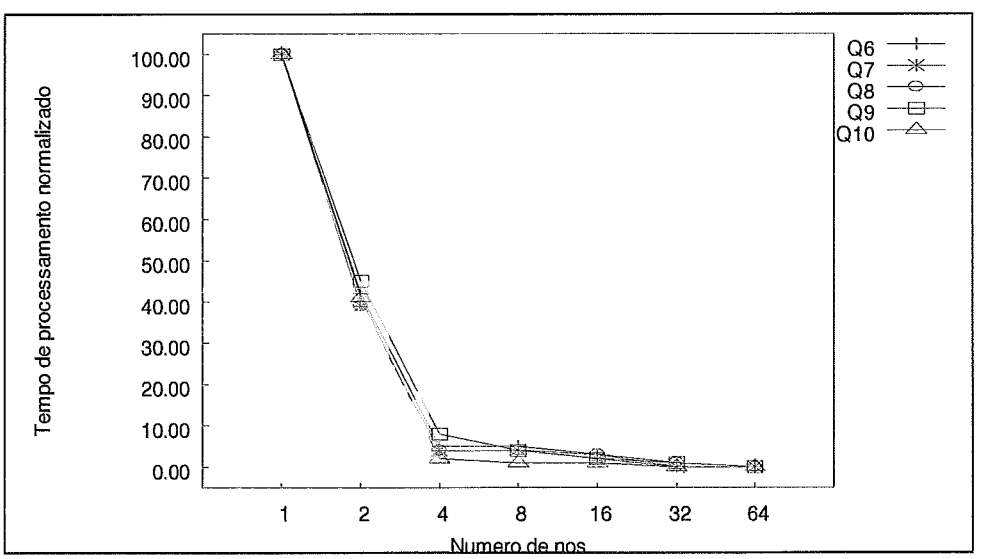


Figura 17: Tempos de execução normalizados por número de nós - Consultas Q6 a Q10

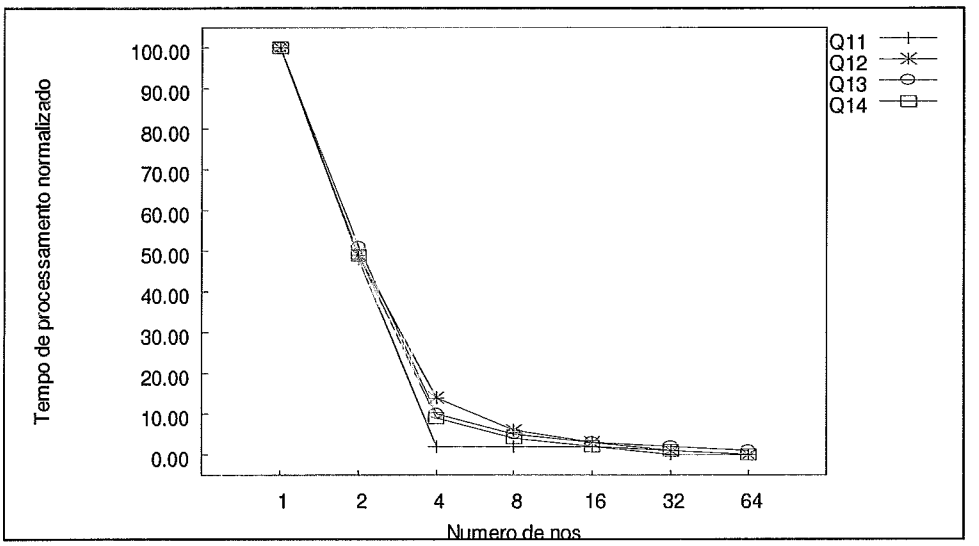


Figura 18: Tempos de execução normalizados por número de nós - Consultas Q11 a Q14

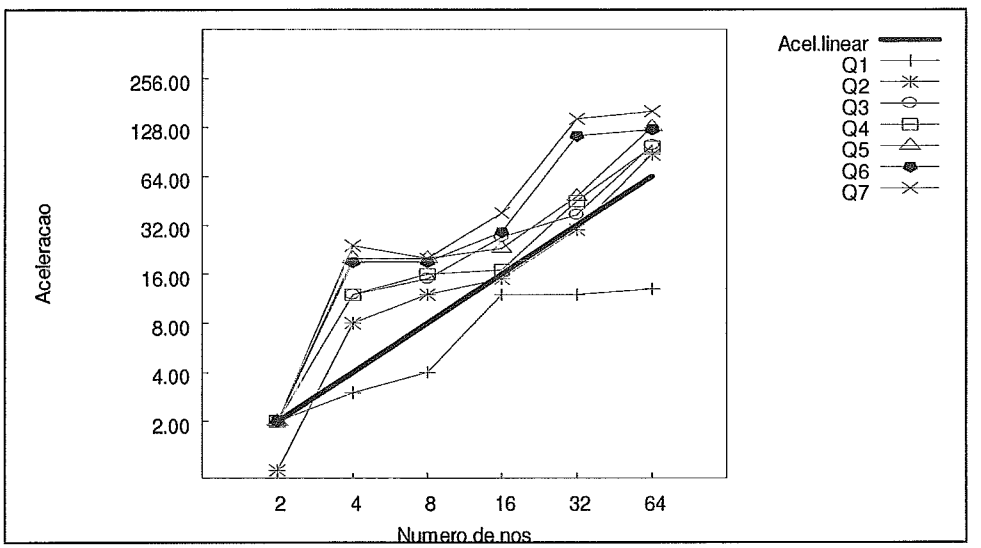


Figura 19: Aceleração super-linear - Consultas Q1 a Q7

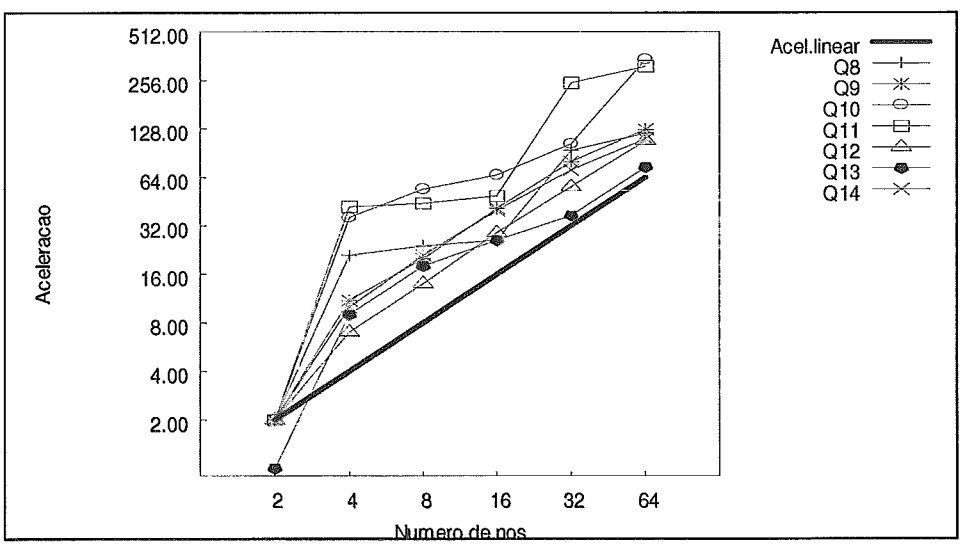


Figura 20: Aceleração super-linear - Consultas Q8 a Q14

5.2.1.2 Experimentos com concorrência

O processamento concorrente de consultas não é uma característica típica de aplicações OLAP, sendo baixo o número de consultas concorrentes. No entanto, conforme análise dos registros de utilização do BME na seção 4.2.4, em que verificamos a ocorrência de consultas realizadas concorrentemente no BME, torna-se essencial avaliar o comportamento do processamento concorrente no ParGRES.

Para simular a concorrência de uma aplicação OLAP, utilizamos processos concorrentes (linhas de execução diferentes) submetendo lotes de consultas para representar usuários dessa aplicação. Esses lotes de consultas são compostos por todas as consultas utilizadas nos experimentos ordenadas de diversas formas. Utilizamos as informações contidas nos registros de execução do BME para definir o número de lotes a serem processados com concorrência e a ordenação das consultas nesses lotes. Os registros de utilização do BME foram apresentados na seção 4.2.4. Com base nessas informações, foram gerados 4 lotes diferentes de consultas, mantendo uma sequência típica de execuções. Observando as 14 consultas escolhidas, podemos achar uma sequência lógica entre 2 ou 3 consultas. Por exemplo, as consultas `Q2`, `Q5` e `Q9` são consultas sobre o número de pessoas em uma dada situação. `Q3` e `Q6` sobre a média da idade das pessoas, onde `Q6` refina os resultados que podem ser obtidos em `Q3`. As consultas `Q7` e `Q8` são consultas sobre o tempo de estudo das pessoas, com diferentes filtros espaciais. E `Q10` e `Q11` são consultas sobre o tempo de estudo das pessoas segundo uma dada informação, com diferentes filtros espaciais.

Vale lembrar que no BME um usuário realiza em média 8 consultas consecutivas e em média 3,7 consultas concorrentes. Então, foram definidos 4 lotes de consultas (simulando 4 usuários executando consultas concorrentemente) e cada lote é composto por todas as 14 consultas do experimento anterior, com as seguintes sequências:

```
L1 = {Q12, Q4, Q2, Q5, Q9, Q13, Q10, Q11, Q7, Q8, Q3, Q6, Q14, Q1}
L2 = {Q1, Q10, Q11, Q12, Q7, Q8, Q13, Q3, Q6, Q14, Q2, Q5, Q9, Q4}
L3 = {Q14, Q13, Q7, Q8, Q12, Q1, Q3, Q6, Q4, Q10, Q11, Q2, Q5, Q9}
L4 = {Q4, Q12, Q3, Q6, Q14, Q2, Q5, Q9, Q1, Q13, Q10, Q11, Q7, Q8}
```

Ao definir a sequência de consultas de cada lote, tomamos o cuidado de não permitir que as consultas coincidissem nas mesmas posições dos lotes, ou seja, em cada posição dos lotes as consultas são diferentes. Por exemplo, na posição 1 dos lotes temos as consultas `Q12`, `Q1`, `Q14` e `Q4`; na posição 2 temos as consultas `Q4`, `Q10`, `Q13` e `Q12`; e

assim sucessivamente; as consultas não se repetem na mesma posição. Assim, reduzimos a possibilidade de uma consulta em um lote tirar proveito dos dados em memória de outro lote sendo executado.

É importante salientar que a realização desse experimento tem como objetivo avaliar o comportamento do processamento paralelo inter e intra-consultas do ParGRES em um cenário extremo de utilização.

5.2.1.2.1 Teste de força

A primeira bateria de execuções é denominada Teste de Força e permite avaliar o tempo de processamento de um mesmo lote, sem concorrência, em diferentes configurações de agrupamento de BD. Ou seja, qual o tempo gasto por um usuário submetendo uma sequência de consultas OLAP com diferentes números de nós.

Os lotes foram processados cinco vezes, sem concorrência e sequencialmente, com 1, 4, 8, 16, 32 e 64 nós e o tempo de execução total de cada lote foi coletado. Em seguida, calculamos a média entre os tempos de cada lote em cada configuração. Mais uma vez, detectamos uma variação entre os tempos coletados. Analisamos essa variação de tempo calculando o desvio padrão dos tempos em relação à média. Todos os tempos coletados (100%) estão entre dois desvios padrões da média, nos levando a ter uma distribuição normal dos tempos de processamento de cada consulta por número de nós. Portanto, utilizamos a média entre os tempos de processamentos como tempo final de processamento. A variação entre os tempos de execução de cada lote por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada lote por número de nós.

Resultados

Apesar de serem formados pelas mesmas consultas, os lotes apresentaram diferentes tempos de processamento, o que pode ser justificado pela sequência em que as consultas foram organizadas dentro de cada um. Ao ser executada, uma consulta pode tirar proveito dos dados em memória da consulta anterior, visto que algumas consultas diferentes acessam os mesmos dados, reduzindo o acesso a disco e o tempo de processamento. O tempo médio de processamento de um lote é de 1 hora e 44 minutos. Com configuração de 4 nós, ocorre uma redução significativa no tempo de processamento de todos os lotes, para aproximadamente 7,5 minutos. Com

configurações superiores, como 8, 16, 32 e 64 nós, embora seja menor, ainda assim há redução do tempo em todos os casos. Com 32 nós, um lote pode ser processado em 1,1 minuto, e com 64 nós, em apenas 41,7 segundos. Os tempos médios de processamento de cada lote podem ser visualizados na Tabela 10.

Tabela 10: Tempo médio de execução dos lotes (em segundos) por número de nós

	Número de nós					
Lotes	1	4	8	16	32	64
L1	6.290,15	503,33	234,90	127,90	73,56	43,46
L2	6.001,06	435,97	216,37	117,00	66,10	41,65
L3	6.343,93	430,79	216,77	114,93	64,19	40,89
L4	6.323,81	431,46	215,93	114,50	64,09	40,81

5.2.1.2.2 Teste de carga

A segunda bateria de execuções é denominada Teste de Carga e permite avaliar o tempo de processamento de diversos lotes concorrentemente, em diferentes configurações de agrupamento de BD. Ou seja, qual o tempo gasto por diversos usuários submetendo diferentes sequências de consultas OLAP simultaneamente com diferentes números de nós.

Neste experimento foram utilizados os mesmos lotes do Teste de Força (`L1`, `L2`, `L3` e `L4`), simulando 4 usuários conectados e realizando consultas, em diferentes configurações de agrupamento. Os lotes foram executados cinco vezes, concorrentemente, através de processos (linhas de execução) diferentes com 4, 8, 16, 32 e 64 nós. O tempo de execução total de cada lote foi coletado. Em seguida, calculamos a média entre os tempos de cada lote em cada configuração. Mais uma vez, detectamos uma variação entre os tempos coletados. Analisamos essa variação de tempo calculando o desvio padrão dos tempos em relação à média. Todos os tempos coletados (100%) estão entre dois desvios padrões da média, nos levando a ter uma distribuição normal dos tempos de processamento de cada consulta por número de nós. Portanto, utilizamos a média entre os tempos de processamentos como tempo final de processamento. A variação entre os tempos de execução de cada lote por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada lote por número de nós.

Resultados

Todos os lotes de consultas foram processados utilizando paralelismo inter e intra-consulta. A Tabela 11 apresenta os tempos médios de processamento de cada lote em cada configuração do agrupamento de banco de dados. Os tempos médios são apresentados em segundos.

Tabela 11: Tempo médio de execução dos lotes concorrentes (em segundos) por número de nós

	Número de nós				
Lotes	4	8	16	32	64
L1	599,21	280,73	155,46	94,50	81,55
L2	588,73	284,10	150,74	91,96	85,94
L3	582,86	280,48	154,22	89,89	80,92
L4	580,54	281,52	152,07	91,69	79,71

Essa tabela mostra que os resultados obtidos nesse experimento foram excelentes porque todos os lotes apresentaram redução no tempo de processamento concorrente à medida que se aumentou o número de nós. Para melhor visualização da redução dos tempos, a Figura 21 apresenta o gráfico com os tempos de execução normalizados.

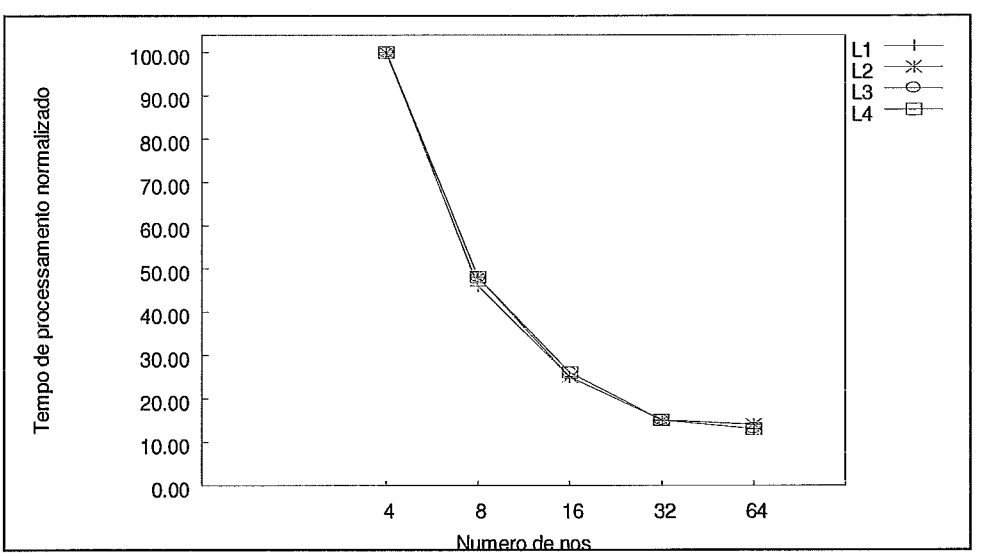


Figura 21: Tempos de execução normalizados por número de nós – Lotes L1 a L4 – **Teste de carga**

Os tempos de processamento de cada lote foram bem parecidos. Com configurações de 8 e 16 nós, houve uma redução significativa no tempo de processamento. Com 32 e 64 nós, embora menor, também ocorre redução no tempo de processamento. Quando executado isoladamente, o lote `L1` demora 8,4 minutos em um ambiente com 4 nós. Com este mesmo número de nós, o processamento concorrente de

L1 com outros lotes demora 10 minutos, ou seja, houve um aumento de apenas 19% no tempo de processamento. Com configurações de 8 e 16 nós, esse aumento é de 19,5% e 21,5%, respectivamente, com 32 nós o aumento é de 28,5%. Com 64 nós, o aumento é de 87,6%, ou seja, o tempo de processamento concorrente dos lotes dobrou quando comparado com o tempo de processamento isolado de cada um deles.

Comparando os tempos de execução dos lotes com concorrência e seus respectivos tempos quando executados isoladamente, observamos que os tempos obtidos com concorrência foram quase sempre muito bons. Mesmo levando mais tempo para serem processados em um ambiente com concorrência, percebemos ganhos nos tempos de processamento dos lotes à medida que novos nós foram adicionados. Assim, podemos concluir que o ParGRES teve um bom comportamento nesse cenário de concorrência em sistemas OLAP, mantendo o desempenho no processamento de consultas paralelas já observado no processamento de consultas sem concorrência.

Na seção 5.2.3, descrevemos um experimento realizado com o objetivo de analisar mais detalhadamente os resultados obtidos com 64 nós e explicar o aumento no tempo de processamento dos lotes com concorrência quando comparados com os tempos sem concorrência. Na mesma seção também explicamos a baixa redução de tempo quando aumentamos o número de nós de 32 para 64.

5.2.2 Experimentos com Replicação Parcial

Esses experimentos têm como finalidade avaliar o desempenho do processamento de consultas utilizando o ParGRES com replicação parcial da base de dados AmCD2000 nos 64 nós que compõem o agrupamento de banco de dados. O projeto de fragmentação foi descrito na seção 3.1.2 do capítulo 3. Em relação ao número de réplicas, nossos experimentos foram realizados utilizando uma e duas réplicas, que foram distribuídas utilizando espalhamento encadeado, descrito na seção 3.1. Além dos tempos de execução, coletamos o número de mensagens de oferta e de aceite de ajuda trocadas, para as consultas Q9 e Q12.

5.2.2.1 Experimentos com consultas isoladas

Esses experimentos foram realizados para avaliar o desempenho do processamento individual de consultas em diferentes configurações de agrupamento de banco de dados (4, 8, 16, 32 e 64 nós), com base parcialmente replicada, utilizando uma

e duas réplicas. Experimentos com 1 e 2 nós não foram realizados, pois nestas configurações não é possível montar um ambiente distribuído significativo. Para construir os gráficos e oferecer uma melhor visualização dos resultados utilizamos os tempos obtidos (para 1 e 2 nós) dos experimentos com base totalmente replicada.

As 14 consultas foram executadas cinco vezes em sequência e seus tempos coletados. Apesar da variação entre os tempos coletados, temos uma distribuição normal. Portanto, utilizamos a média entre os tempos de processamentos como tempo final. A variação entre os tempos de execução de cada consulta por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada consulta por número de nós.

Resultados (1 e 2 réplicas)

Todas as consultas foram processadas utilizando paralelismo intra-consulta. A Tabela 12 apresenta os tempos médios de processamento de cada consulta em cada configuração do agrupamento de banco de dados, com uma e duas réplicas. Os tempos médios são apresentados em segundos.

Tabela 12: Tempo médio de execução das consultas (em segundos) por número de nós e réplicas

	1R					2R				
	Número de nós									
Consultas	4	8	16	32	64	4	8	16	32	64
Q1	4,25	2,11	1,22	1,01	2,50	5,57	2,16	1,28	1,20	2,93
Q2	13,55	7,03	3,72	2,32	4,21	13,86	7,09	3,76	2,37	4,40
Q3	16,27	8,30	4,28	2,42	4,11	16,42	8,36	4,38	2,47	4,30
Q4	14,66	7,51	4,05	2,71	3,59	15,10	7,56	4,82	2,77	3,64
Q5	6,93	3,68	1,99	1,33	3,59	7,46	3,72	2,20	1,39	3,64
Q6	9,72	5,04	2,63	1,65	1,91	9,74	5,32	2,75	1,71	3,49
Q7	5,49	2,90	1,85	1,39	1,51	5,61	2,99	2,02	1,46	3,35
Q8	9,29	4,90	2,68	1,68	2,30	9,45	5,10	2,72	1,74	3,79
Q9	39,72	20,2	10,37	5,65	5,71	40,03	20,30	10,6	5,81	7,06
Q10	10,60	5,79	3,19	1,97	3,44	10,78	5,87	3,23	2,15	3,50
Q11	11,41	6,01	3,55	2,13	2,30	11,81	6,25	3,74	2,21	4,61
Q12	168,32	81,81	41,14	21,44	13,49	173,64	82,09	41,6	21,55	13,57
Q13	21,58	11,06	6,12	3,79	5,09	23,06	11,3	6,18	3,86	5,33
Q14	94,59	48,24	25,98	14,52	11,58	95,6	48,47	26,02	15,10	12,09

Os resultados obtidos nesse experimento foram muito bons até 32 nós, porque todas as consultas apresentaram redução no tempo de processamento à medida que se aumentou o número de nós. No entanto, com 64 nós houve um ligeiro aumento no

tempo de processamento das consultas, com exceção de `Q12` e `Q14`, que apresentaram redução nos tempos em todas as configurações de agrupamento, tanto com 1 quanto 2 réplicas. A consulta `Q1` apresentou o maior aumento de tempo nesta configuração.

Da Figura 22 à Figura 27, apresentamos os gráficos com os tempos de execução normalizados para melhor visualização dos resultados. Os tempos normalizados foram obtidos dividindo o tempo de execução de cada consulta pelo seu maior tempo de execução, no nosso caso, o tempo sequencial da consulta (tempo de processamento de uma consulta com um nó).

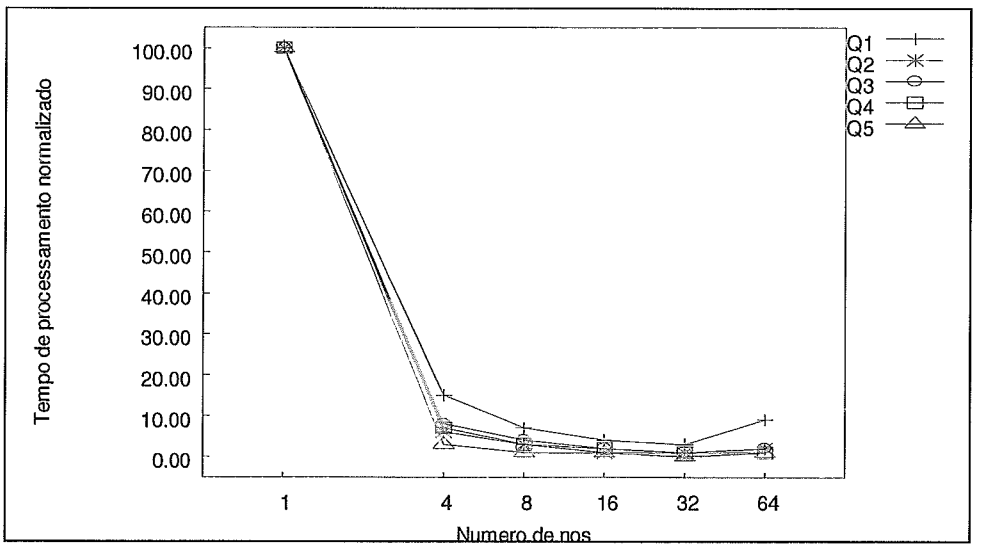


Figura 22: Tempos de execução normalizados por número de nós – Consultas Q1 a Q5 – **1 réplica**

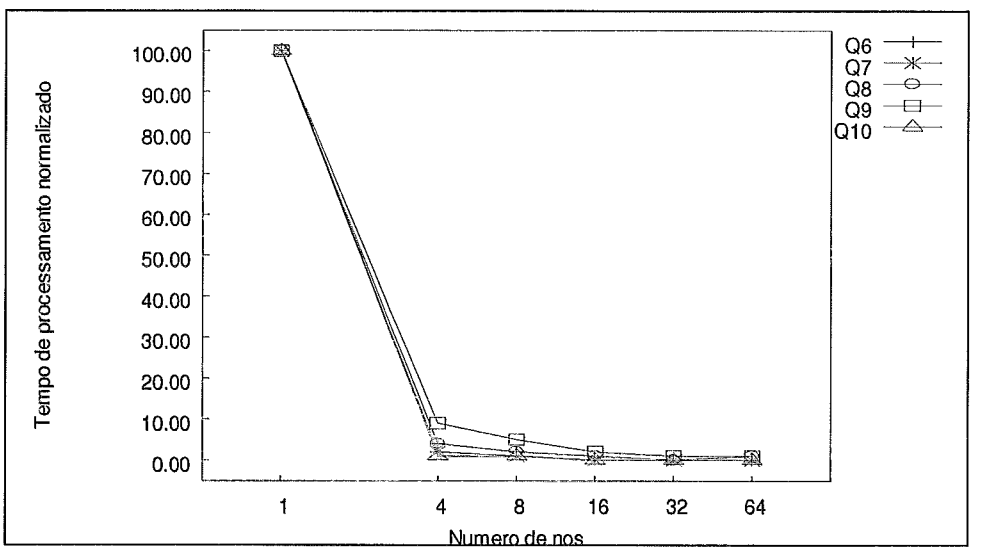


Figura 23: Tempos de execução normalizados por número de nós – Consultas Q6 a Q10 – **1 réplica**

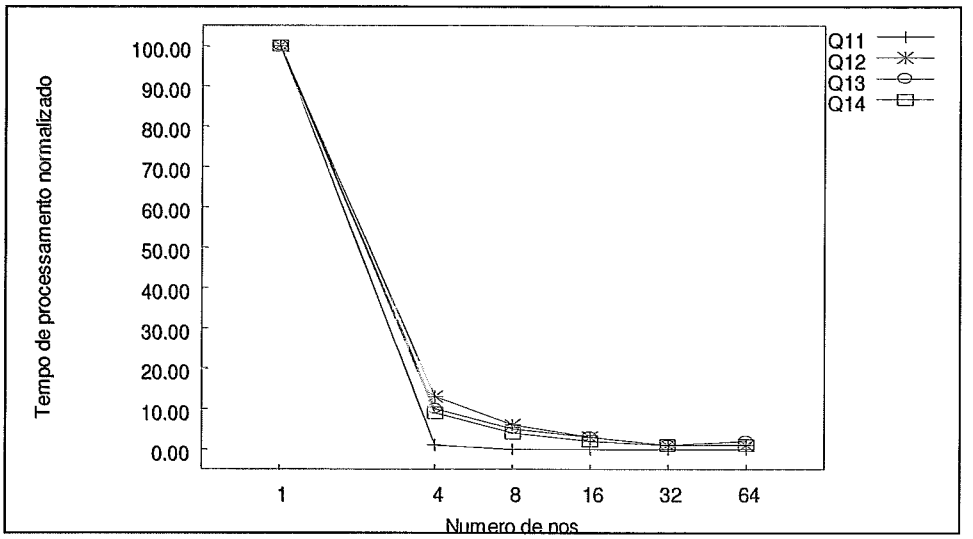


Figura 24: Tempos de execução normalizados por número de nós – Consultas Q11 a Q14 – **1 réplica**

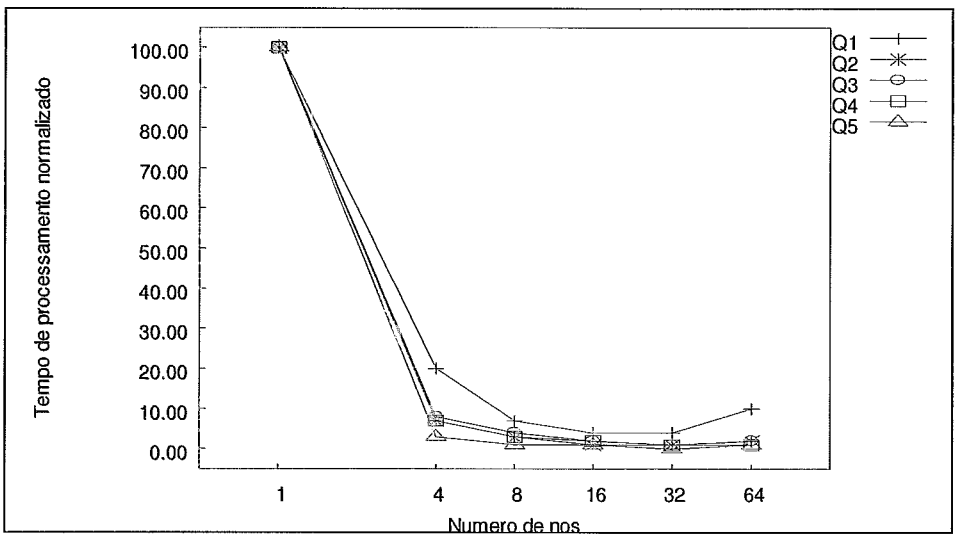


Figura 25: Tempos de execução normalizados por número de nós – Consultas Q1 a Q5 – **2 réplicas**

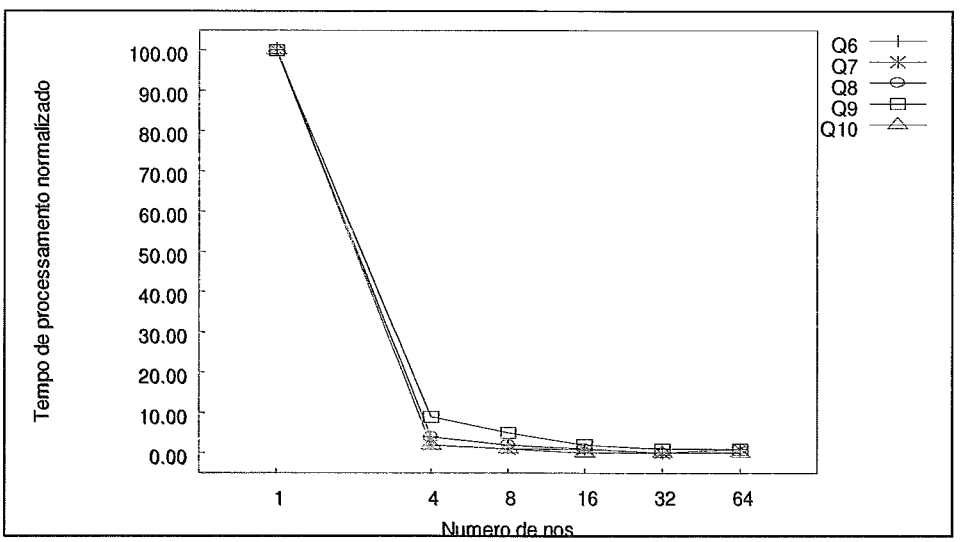


Figura 26: Tempos de execução normalizados por número de nós – Consultas Q6 a Q10 – **2 réplicas**

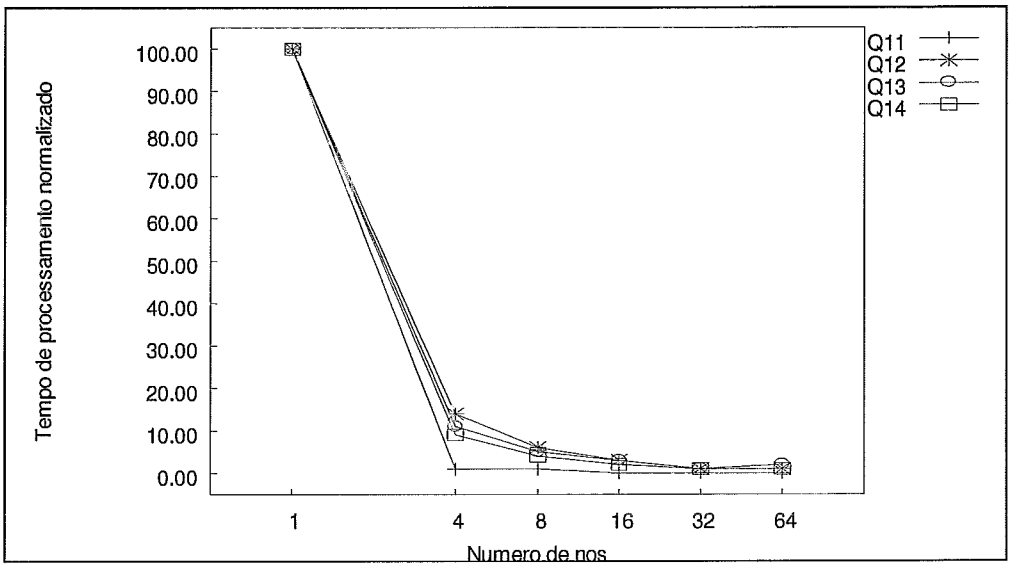


Figura 27: Tempos de execução normalizados por número de nós – Consultas Q11 a Q14 – **2 réplicas**

Da Figura 28 a Figura 31, apresentamos os gráficos com as curvas de aceleração super-linear obtidas nesses experimentos. Para facilitar a visualização da aceleração super-linear, foi plotada uma linha reta e grossa, representando a curva de aceleração linear.

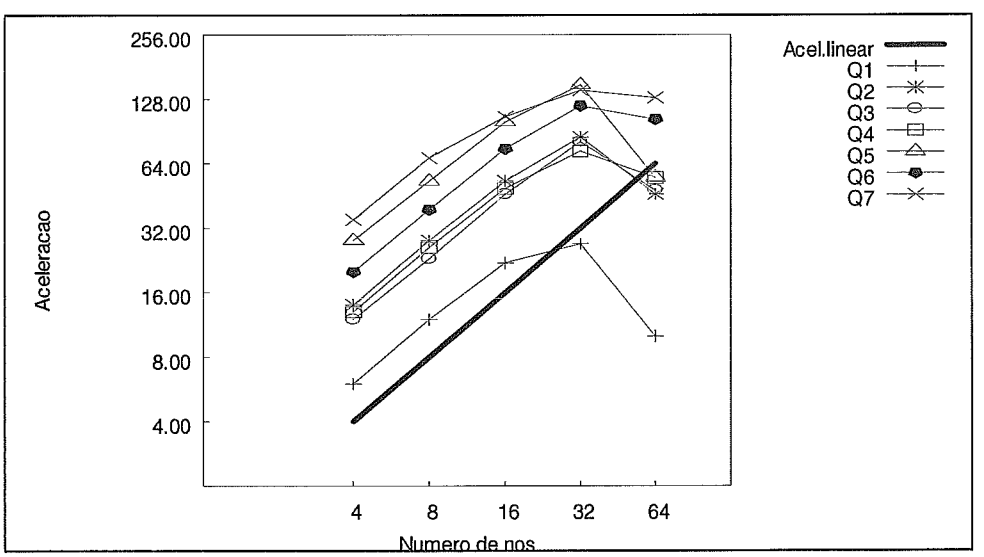


Figura 28: Aceleração super-linear – Consultas Q1 a Q7 – **1 réplica**

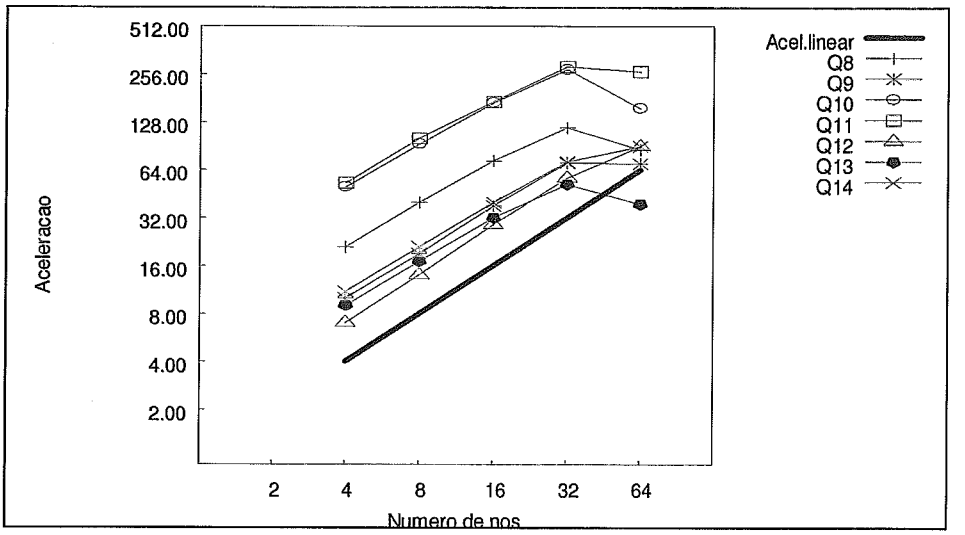


Figura 29: Aceleração super-linear – Consultas Q8 a Q14 – **1 réplica**

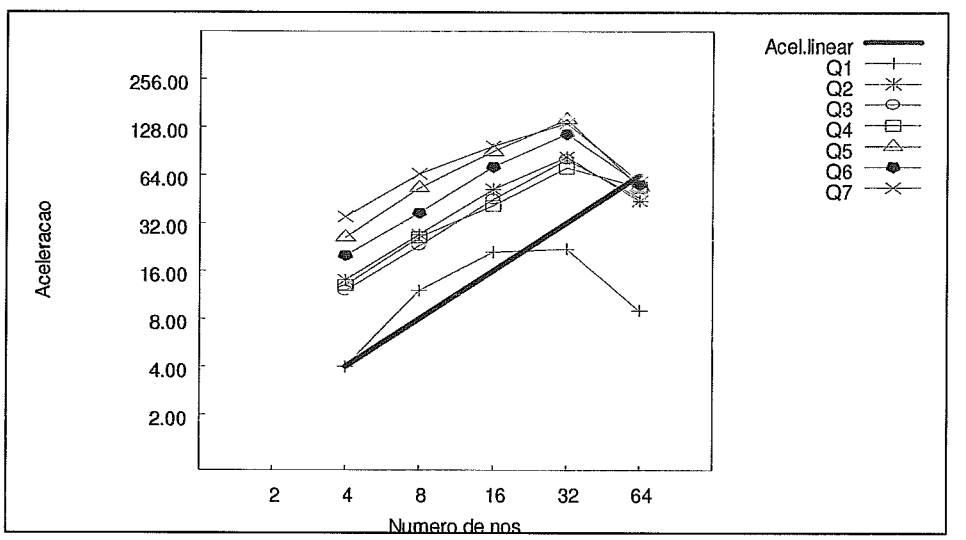


Figura 30: Aceleração super-linear – Consultas Q1 a Q7 – **2 réplicas**

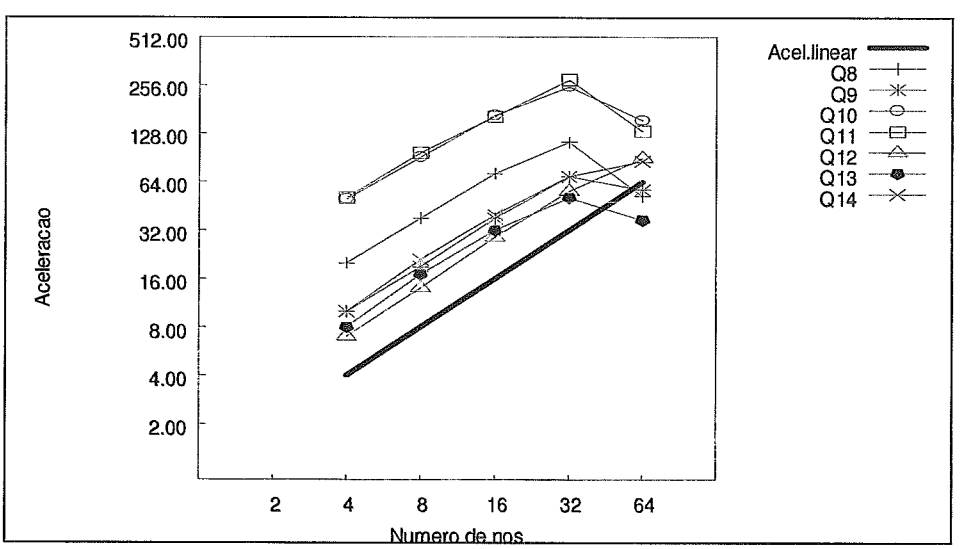


Figura 31: Aceleração super-linear – Consultas Q8 a Q14 – **2 réplicas**

Em bases parcialmente replicadas com 1 réplica, todas as consultas obtiveram aceleração super-linear em todas as configurações do agrupamento até 32 nós. Com 64 nós, as consultas `Q2`, `Q3`, `Q4`, `Q5` e `Q13` obtiveram aceleração sub-linear; as demais obtiveram aceleração super-linear mesmo com o aumento de tempo em relação ao número de nós anterior. A consulta `Q1` obteve aceleração sub-linear com 32 e 64 nós. Com configurações de 16 e 32 nós, a redução no tempo de processamento não é significativa como em configurações de 4 e 8 nós, no entanto, foram as maiores acelerações super-lineares obtidas.

O comportamento acima descrito se repete em bases com 2 réplicas, mais uma vez com exceção da `Q1`, que obteve aceleração linear com 4 nós e sub-linear, com 32 e 64 nós. Com 64 nós, apenas as consultas `Q10`, `Q11`, `Q12` e `Q14` obtiveram aceleração super-linear, mesmo com o ligeiro aumento de tempo.

Observamos que as consultas tiveram seu tempo de processamento afetado pelo custo envolvido na composição de resultados, que tende a ser maior com um número elevado de nós. Na seção 5.2.3 descrevemos um experimento adicional que investiga o tempo de processamento das consultas e o tempo de composição dos resultados.

Os tempos de processamento das consultas em base parcial com 2 réplicas são maiores que os tempos com 1 réplica. Esse comportamento é explicado na seção 5.3, na qual apresentamos dados sobre as mensagens de oferta de ajuda. Podemos concluir que ao aumentar o número de réplicas aumenta-se o número de mensagens trocadas, e haver mais oferta de ajuda não quer dizer que exista mais ajuda efetiva. Ou seja, há mais consumo de espaço em disco com menos benefício.

5.2.2.2 Experimentos com concorrência

Em experimentos com base parcialmente replicada (com uma e duas réplicas) também realizamos processamento concorrente de consultas para avaliar o comportamento do ParGRES. Para simular a concorrência, utilizamos processos (linhas de execução) concorrentes submetendo lotes de consultas para representar usuários dessa aplicação. Os lotes de consultas utilizados foram os mesmos utilizados no experimento descrito na seção 5.2.1.2.

5.2.2.2.1 Teste de força

Os lotes (`L1, L2, L3 e L4`) foram processados cinco vezes, sem concorrência e sequencialmente, com 4, 8, 16, 32 e 64 nós, com bases parcialmente replicadas, utilizando uma e duas réplicas. O tempo de execução total de cada lote foi coletado e, depois, foi calculada a média entre os tempos de cada lote em cada configuração. Detectamos uma variação entre os tempos coletados e analisamos essa variação calculando o desvio padrão dos tempos em relação à média. Todos os tempos coletados (100%) estão entre dois desvios padrões da média, nos levando a ter uma distribuição normal dos tempos de processamento de cada consulta por número de nós. Portanto, utilizamos a média entre os tempos de processamentos como tempo final de processamento. A variação entre os tempos de execução de cada lote por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada lote por número de nós.

Resultados (1 e 2 réplicas)

Todos os lotes de consultas foram processados utilizando paralelismo intra-consulta. A Tabela 13 apresenta os tempos médios de processamento de cada lote em cada configuração do agrupamento de banco de dados, utilizando uma e duas réplicas. Os tempos médios são apresentados em segundos.

Com 4 nós, o tempo médio de processamento de um lote é de 7,3 minutos com 1 réplica e 7,4 minutos com 2 réplicas. Com 32 nós esse tempo reduz para aproximadamente 1,1 minutos (com 1 e 2 réplicas) e para 1,2 minutos e 1,3 minutos com 1 e 2 réplicas, respectivamente, com 64 nós. Em configurações de até 32 nós houve uma redução significativa no tempo médio de processamento de um lote, tanto com 1 quanto com 2 réplicas,. Com 64 nós observamos o mesmo comportamento dos experimentos com consultas isoladas, em que houve um ligeiro aumento no tempo de processamento.

Tabela 13: Tempo médio de execução dos lotes (em segundos) por número de nós e réplicas

	1R					2R				
	Número de nós									
Consultas	**4**	**8**	**16**	**32**	**64**	**4**	**8**	**16**	**32**	**64**
L1	479,55	212,29	127,59	65,81	76,57	486,55	234,13	116,45	67,60	79,56
L2	424,63	212,13	113,66	64,39	73,25	433,54	215,57	113,92	65,99	76,75
L3	427,06	211,91	112,82	62,48	74,27	429,92	214,20	112,68	63,66	76,39
L4	426,94	212,03	112,84	62,51	74,30	431,04	215,33	113,01	63,91	76,95

5.2.2.2.2 Teste de carga

Os lotes (`L1, L2, L3 e L4`) foram processados simulando 4 usuários conectados e realizando consultas, em diferentes configurações de agrupamento (4, 8, 16, 32 e 64 nós), com 1 e 2 réplicas. Os lotes foram executados cinco vezes, concorrentemente, através de processos (linhas de execução) diferentes. O tempo de execução total de cada lote foi coletado e, em seguida, foi calculada a média entre os tempos. Mais uma vez, foi detectada uma variação entre os tempos coletados e ao analisar essa variação através do cálculo do desvio padrão dos tempos em relação à média, todos os tempos coletados (100%) mantiveram-se entre dois desvios padrões da média. Portanto, utilizamos a média entre os tempos de processamentos como tempo final de processamento. A variação entre os tempos de execução de cada lote por configuração de agrupamento pode ser visualizada nos gráficos do Anexo D, onde todos os tempos de execução foram plotados, inclusive os gráficos com barras de erro, que apresentam os desvios padrões da média de cada lote por número de nós.

Resultados (1 e 2 réplicas)

Todos os lotes de consultas foram processados utilizando paralelismo inter e intra-consulta. Como nos experimentos com concorrência em bases totalmente replicadas, observamos o mesmo comportamento dos tempos de processamento, em que estes são maiores que os tempos de processamento dos lotes executados isoladamente. A Tabela 14 apresenta os tempos médios de processamento de cada lote em cada configuração do agrupamento de banco de dados, com uma e duas réplicas. Os tempos médios são apresentados em segundos. Em todas as configurações de nós no agrupamento até 32 nós, os tempos diminuíram à medida que os nós foram adicionados. Com 64 nós, houve um ligeiro aumento nos tempos, perceptível nos gráficos da Figura 32 e da Figura 33.

Tabela 14: Tempo médio de execução dos lotes concorrentes (em segundos) por número de nós

	1R					2R				
	Número de nós									
Consultas	4	8	16	32	64	4	8	16	32	64
L1	717,39	301,48	160,33	135,24	217,23	697,56	285,75	156,08	130,70	215,80
L2	730,67	307,11	167,55	139,74	215,12	725,18	285,51	151,89	134,08	214,89
L3	741,45	296,38	160,36	126,21	217,58	714,53	286,91	155,05	125,07	216,27
L4	731,21	307,37	168,65	138,47	217,98	697,39	284,87	152,81	134,63	217,07

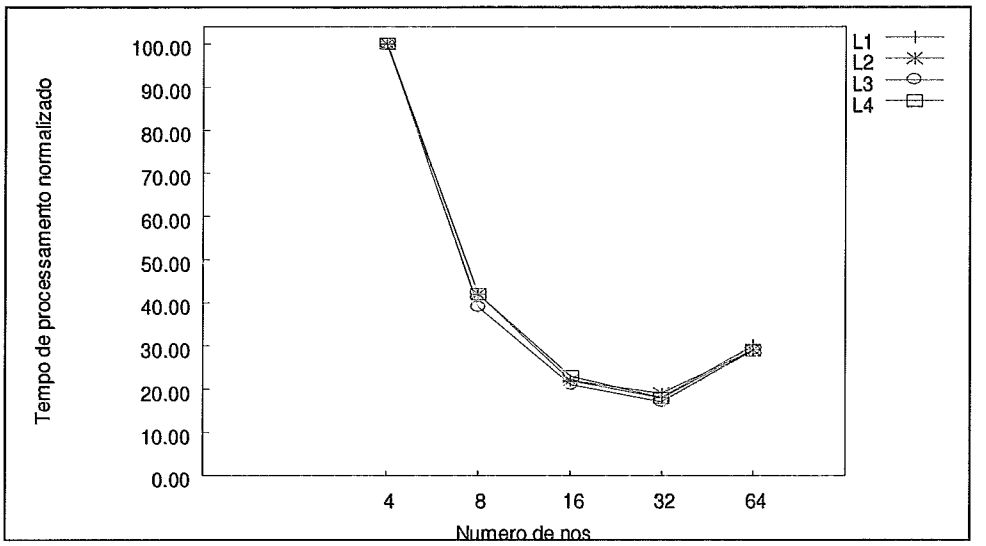


Figura 32: Tempos de execução normalizados por número de nós – Lotes L1 a L4 – **Teste de carga – 1 réplica**

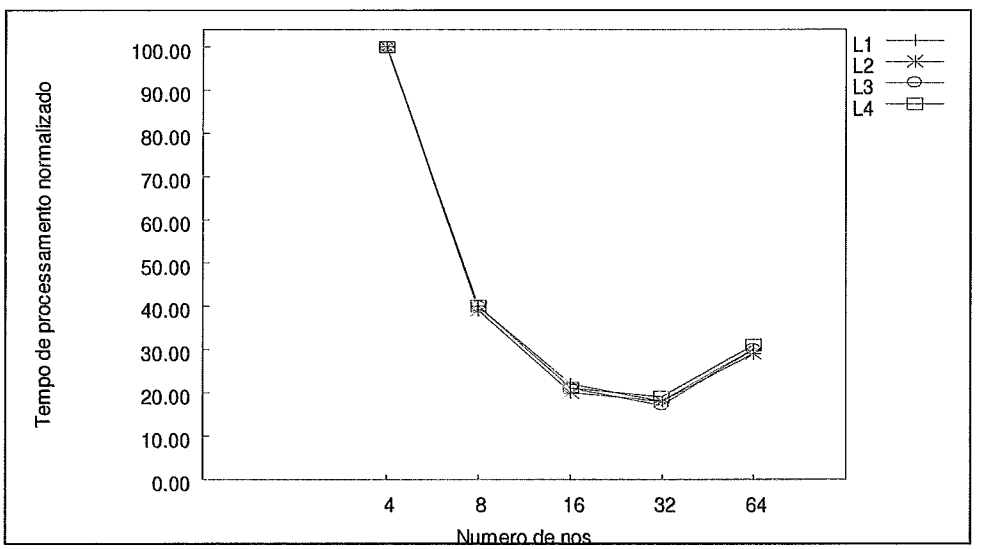


Figura 33: Tempos de execução normalizados por número de nós – Lotes L1 a L4 – **Teste de carga – 2 réplicas**

5.2.3 Experimentos Adicionais

Como apresentado nas seções anteriores, a redução nos tempos de processamento das consultas em configurações de agrupamento com 64 nós, em bases totalmente replicadas foi muito pequena, e houve um ligeiro aumento nos tempos de processamento de grande parte das consultas, com 64 nós, em bases parcialmente replicadas, com 1 e 2 réplicas. A pequena redução e o aumento nos tempos de processamento das consultas com 64 nós, nos motivou a realizar novos experimentos com consultas isoladas em um nível maior de detalhamento a fim de identificar a causa

do comportamento observado e prover uma análise mais adequada dos resultados obtidos. A seguir, descrevemos o experimento adicional realizado.

O experimento foi realizado com 32 e 64 nós em bases totalmente replicadas, e com 32, 40, 48, 56 e 64, em bases parcialmente replicadas, com 1, 4 e 8 réplicas. Apenas as consultas `Q9` e `Q12` foram selecionadas para essa bateria de testes, pois estas possuem características que estão presentes nas demais consultas. Além de terem sido executadas cinco vezes consecutivas, com seus tempos coletados e médias calculadas, coletamos também o tempo de processamento de cada sub-consulta em cada nó do agrupamento. Lembrando que até agora foram apresentados apenas os tempos de sala, ou seja, o tempo total de processamento. No tempo total está incluído o tempo gasto pelo ParGRES para organizar os resultados enviados por cada nó e enviar o resultado final para o usuário. O nosso objetivo é medir o tempo gasto na composição dos resultados das consultas e analisar a diferença entre o tempo total de processamento de uma consulta e o tempo individual de processamento de cada sub-consulta (gerada pela fragmentação virtual) da consulta em cada nó. Sabemos que quanto maior o número de nós, maior o custo envolvido na composição dos resultados, ou seja, o número de resultados é proporcional ao número de nós. Essas informações são apresentadas na Tabela 15 e na Tabela 16.

Com 32 nós, em bases total e parcialmente replicadas, a diferença entre o tempo médio total e individual é muito pequena, mantendo o tempo total muito próximo do tempo individual. Com 40 nós, na maioria das vezes já observamos aumento em ambos os tempos, total e individual, comparando esse tempo com a configuração anterior; a diferença entre os tempos mantém-se estável. Com 48 nós, a diferença entre os tempos atinge 42%, mas em grande parte, a diferença é baixa, em torno de 10%. Com 56 e 64 nós, o tempo médio total tem um aumento significativo sobre o tempo individual, fazendo com que o tempo de processamento seja maior do que o obtido com a configuração anterior, e essa diferença atinja até 374%. Embora a consulta `Q12` não tenha apresentado aumento em seu tempo com 64 nós, a diferença entre o tempo médio total e o tempo médio individual também foi alta, tal como a diferença entre os tempos da consulta `Q9`. Em configurações de 64 nós detectamos as maiores diferenças entre o tempo total e o tempo individual.

O custo envolvido na composição de resultados em configurações de agrupamentos maiores que 32 nós se tornou relevante para o tamanho dos experimentos desta dissertação, influenciando no tempo total do processamento das consultas.

Tabela 15: Tempo total x Tempo individual (em segundos) - Q9 e Q12 – **Base totalmente replicada**

Q9	Número de nós	
	32	64
Tempo médio total (TT)	7,64	8,80
Tempo médio individual (LQTs) (TI)	6,41	3,64
Diferença entre TT e TI	1,24	5,16
Q12	**Número de nós**	
	32	64
Tempo médio total (TT)	21,66	15,05
Tempo médio individual (LQTs) (TI)	21,18	11,48
Diferença entre TT e TI	0,47	3,57

Tabela 16: Tempo total x Tempo Individual (em segundos) - Q9 e Q12 – **Base parcialmente replicada**

Nº réplicas	Q9	Número de nós				
		32	40	48	56	64
1R	Tempo médio total (TT)	7,12	11,49	10,07	12,68	9,03
	Tempo médio individual (LQTs) (TI)	5,81	10,25	8,31	7,27	3,48
	Diferença entre TT e TI	1,32	1,23	1,76	5,41	5,55
4R	Tempo médio total (TT)	8,92	10,05	9,99	10,62	9,18
	Tempo médio individual (LQTs) (TI)	7,20	8,47	7,99	6,53	4,51
	Diferença entre TT e TI	1,72	1,57	2,00	4,10	4,68
8R	Tempo médio total (TT)	9,87	10,36	7,10	8,19	10,17
	Tempo médio individual (LQTs) (TI)	8,64	8,79	5,20	3,79	5,35
	Diferença entre TT e TI	1,23	1,57	1,90	4,40	4,82
	Q12	**Número de nós**				
		32	40	48	56	64
1R	Tempo médio total (TT)	22,87	18,63	16,53	16,85	14,58
	Tempo médio individual (LQTs) (TI)	21,66	17,85	15,54	13,55	11,21
	Diferença entre TT e TI	1,21	0,78	1,00	3,30	3,36
4R	Tempo médio total (TT)	21,96	18,18	15,5	16,11	15,91
	Tempo médio individual (LQTs) (TI)	21,46	17,14	14,8	12,77	11,3
	Diferença entre TT e TI	0,51	1,04	0,70	3,34	4,61
8R	Tempo médio total (TT)	21,81	26,69	15,96	16,02	15,05
	Tempo médio individual (LQTs) (TI)	21,35	25,43	14,56	12,96	11,55
	Diferença entre TT e TI	0,46	1,26	1,39	3,06	3,50

5.3 Análise comparativa entre ambientes

Nesta seção realizamos algumas comparações entre os diversos ambientes do experimento.

Mensagens de oferta e de aceite de ajuda *versus* ajuda efetiva

Em todos os experimentos realizados foram coletadas as mensagens enviadas pelos nós referentes à oferta e aceite de ajuda. Na Tabela 17 apresentamos o número de mensagens de oferta de ajuda, de aceite de ajuda e a ajuda efetivamente dada. É

importante salientar que um nó pode aceitar a ajuda oferecida, mas não ser ajudado. Portanto, a ajuda efetivamente dada é aquela em que o nó aceitou a ajuda e foi de fato ajudado por outro nó. Essas informações foram coletadas apenas para as mensagens `Q9` e `Q12`, devido à complexidade de coleta destas estatísticas.

Em bases parcialmente replicadas, com 1 réplica, o número de mensagens de oferta de ajuda é menor e o índice de ajuda efetivamente dada é maior, comparando esse número com 2 réplicas e bases totalmente replicadas. Com 4 e 8 nós, 69,2% e 68%, respectivamente, das mensagens de ajuda foram aceitas e seus respectivos nós foram ajudados. Com 16 e 32 nós, 57,9% e 55,6%, respectivamente, das mensagens de ajuda foram aceitas e a ajuda foi dada. Com 64 nós, a ajuda efetiva foi de 38,5%.

Em bases parcialmente replicadas, com 2 réplicas, o número de mensagens de oferta de ajuda é maior do que o número com 1 réplica e menor do que em bases totalmente replicadas; o índice de ajuda efetivamente dada é menor que a ajuda dada com 1 réplica e bem maior quando comparada com bases totalmente replicadas. Com 4 e 8 nós, 42,9% e 56,3%, respectivamente, das mensagens de ajuda foram aceitas e seus respectivos nós foram ajudados. Com 16 e 32 nós, 37,5% e 35,7%, respectivamente, das mensagens de ajuda foram aceitas e a ajuda foi dada. Com 64 nós, a ajuda efetiva foi de 25,6%.

Em bases totalmente replicadas, o número de mensagens de oferta de ajuda é muito maior e a ajuda efetivamente dada é muito menor. Com 4 e 8 nós, 32,3% e 29,3%, respectivamente, das mensagens de ajuda foram aceitas e seus respectivos nós foram ajudados. Com 16 e 32 nós, 12,5% e 9,1%, respectivamente, das mensagens de ajuda foram aceitas e a ajuda foi dada. Com 64 nós, a ajuda efetiva foi de 3,5%.

Tabela 17: Número de mensagens trocadas entre os nós (em número de mensagens) - Consultas Q9 e Q12

	Q9			Q12		
	Base total	Base parcial		Base total	Base parcial	
		1 réplica	2 réplicas		1 réplica	2 réplicas
4 nós						
Oferta de ajuda	38	8	18	62	13	14
Aceite de ajuda	20	7	14	47	12	9
Ajuda efetivamente dada	8	4	5	20	9	3
8 nós						
Oferta de ajuda	86	25	64	147	22	64
Aceite de ajuda	60	23	50	110	19	58
Ajuda efetivamente dada	20	17	24	43	14	24
16 nós						

Oferta de ajuda	416	30	90	380	38	28
Aceite de ajuda	155	23	75	154	34	113
Ajuda efetivamente dada	52	14	29	40	22	48
32 nós						
Oferta de ajuda	770	52	184	790	72	224
Aceite de ajuda	239	40	135	316	63	185
Ajuda efetivamente dada	44	20	60	72	40	80
64 nós						
Oferta de ajuda	2240	81	198	1581	104	262
Aceite de ajuda	330	50	118	485	72	185
Ajuda efetivamente dada	28	17	35	55	40	67

Podemos observar que em bases parcialmente replicadas com 1 e 2 réplicas, houve redução do número de mensagens trocadas em relação ao número em bases totalmente replicadas. Com 2 réplicas, o número de mensagens enviadas é maior que o número de mensagens enviadas com 1 réplica; e com base totalmente replicada, o número de mensagens é maior que com 2 réplicas. Isso nos permite afirmar que à medida que adicionamos réplicas, seguindo na direção da replicação total, aumenta o número de mensagens de oferta de ajuda e diminui a proporção de ajuda efetivamente dada.

Tempo de processamento de consultas isoladas: base totalmente replicada *versus* base parcialmente replicada

Para permitir uma melhor visualização do desempenho de processamento de consultas ao longo dos experimentos realizados em bases total e parcialmente replicadas, organizamos os tempos médios de processamento (em segundos) das consultas `Q9` e `Q12` na Tabela 18.

Tabela 18: Tempo de processamento de consultas isoladas (em segundos) – Base totalmente replicada x Base parcialmente replicada (1 e 2 réplicas)

	Base totalmente replicada	**Base parcial - 2R**	**Base parcial - 1R**
Consulta	**4 nós**		
Q1	9,00	5,57	4,25
Q2	24,41	13,86	13,55
Q3	16,81	16,42	16,27
Q4	16,89	15,10	14,66
Q5	10,70	7,46	6,93
Q6	11,45	9,74	9,72
Q7	12,33	5,61	5,49
Q8	10,08	9,45	9,29
Q9	41,30	40,03	39,72
Q10	16,18	10,78	10,60

Q11	13,35	11,81	11,41
Q12	174,95	173,64	168,32
Q13	23,15	23,06	21,58
Q14	93,80	95,60	94,59
	8 nós		
Q1	7,86	2,16	2,11
Q2	17,85	7,09	7,03
Q3	13,96	8,36	8,30
Q4	12,88	7,56	7,51
Q5	10,38	3,72	3,68
Q6	11,42	5,32	5,04
Q7	10,55	2,99	2,90
Q8	9,01	5,10	4,90
Q9	22,52	20,30	20,20
Q10	10,86	5,87	5,79
Q11	12,74	6,25	6,01
Q12	84,45	82,09	81,81
Q13	11,68	11,30	11,06
Q14	48,08	48,47	48,24
	16 nós		
Q1	2,61	1,28	1,22
Q2	13,70	3,76	3,72
Q3	7,94	4,38	4,28
Q4	12,36	4,82	4,05
Q5	9,12	2,20	1,99
Q6	7,44	2,75	2,63
Q7	5,60	2,02	1,85
Q8	8,35	2,72	2,68
Q9	11,03	10,60	10,37
Q10	8,86	3,23	3,19
Q11	11,54	3,74	3,55
Q12	42,25	41,60	41,14
Q13	8,15	6,18	6,12
Q14	25,50	26,02	25,98
	32 nós		
Q1	2,50	1,20	1,01
Q2	7,18	2,37	2,32
Q3	5,78	2,47	2,42
Q4	4,79	2,77	2,71
Q5	4,47	1,39	1,33
Q6	1,91	1,71	1,65
Q7	1,51	1,46	1,39
Q8	2,30	1,74	1,68
Q9	5,71	5,81	5,65
Q10	5,70	2,15	1,97
Q11	2,30	2,21	2,13
Q12	21,91	21,55	21,44
Q13	5,83	3,86	3,79
Q14	14,45	15,10	14,52
	64 nós		
Q1	2,36	2,93	2,50
Q2	2,50	4,40	4,21
Q3	2,19	4,30	4,11

Q4	2,24	3,64	3,59
Q5	1,69	3,64	3,59
Q6	1,75	3,49	1,91
Q7	1,35	3,35	1,51
Q8	1,81	3,79	2,30
Q9	3,60	7,06	5,71
Q10	1,70	3,50	3,44
Q11	1,82	4,61	2,30
Q12	11,43	13,57	13,49
Q13	2,94	5,33	5,09
Q14	9,20	12,09	11,58

Com base totalmente replicada, o tempo de processamento foi maior que com base parcial, com 1 e 2 réplicas. À medida que o número de réplicas diminuiu, o tempo de processamento foi menor. Esse comportamento foi observado em configurações de agrupamento de até 32 nós. Com 4 nós, a redução no tempo de processamento chegou a 55%; com 8 nós, 73%; com 16 nós, 78%; e com 32 nós, redução de até 70%. A consulta `Q14` apresentou um comportamento diferenciado das demais em todas as configurações do agrupamento, pois os seus tempos aumentaram à medida que o número de réplicas diminuiu. Esse comportamento pode ser justificado pelas características da consulta, que possui junções entre todas as tabelas de fatos e 6 tabelas de dimensão, além de possuir um operador de disjunção (OR) em seu predicado.

Com 64 nós, o menor tempo de processamento foi obtido com base totalmente replicada e o maior tempo com 2 réplicas. Comparando os tempos entre base totalmente replicada e 1 réplica, o aumento chegou a 112%, e comparando os tempos entre base total e 2 réplicas, o aumento chegou a 154%.

Analisando os tempos de processamento e a troca de mensagens de oferta e aceite de ajuda conjuntamente, podemos concluir que o envio de mensagens para um grupo maior de nós provoca uma degradação no desempenho de processamento, pois há o custo envolvido em receber a ajuda, verificar se ela é necessária e em responder (aceitando ou não) a oferta. Nesses experimentos detectamos que o desbalanceamento de carga foi muito pequeno. Logo, quanto maior o número de réplicas, maior o número de mensagens enviadas, e maior o tempo de processamento, principalmente em cenários de baixo desbalanceamento de carga.

Tempo de processamento de lotes concorrentes: base totalmente replicada *versus* base parcialmente replicada

Para permitir uma melhor visualização no desempenho do processamento dos lotes concorrentes ao longo dos experimentos realizados em bases total e parcialmente replicadas, organizamos os tempos médios de processamento (em segundos) desses lotes na Tabela 19.

Com base totalmente replicada, os tempos de processamento foram menores que com bases parciais, com 1 e 2 réplicas. À medida que o número de réplicas diminuiu, o tempo de processamento aumentou. Esse comportamento foi observado em todas as configurações de agrupamento. Com 4 nós, o aumento no tempo de processamento chegou a 27%; com 8 nós, 9%; com 16 nós, 11%; com 32 nós, 52%; e com 64 nós, aumento de até 173%. O maior aumento de tempo foi observado em configurações de 64 nós.

Tabela 19: Tempo de processamento de lotes concorrentes (em segundos) – Base totalmente replicada x Base parcialmente replicada (1 e 2 réplicas)

	Base totalmente replicada	**Base parcial - 2R**	**Base parcial - 1R**
Lote	**4 nós**		
L1	599,21	697,56	717,39
L2	588,73	725,18	730,67
L3	582,86	714,53	741,45
L4	580,54	697,39	731,21
	8 nós		
L1	280,73	285,75	301,48
L2	284,10	285,51	307,11
L3	280,48	286,91	296,38
L4	281,52	284,87	307,37
	16 nós		
L1	155,46	156,08	160,33
L2	150,74	151,89	167,55
L3	154,22	155,05	160,36
L4	152,07	152,81	168,65
	32 nós		
L1	94,50	130,70	135,24
L2	91,96	134,08	139,74
L3	89,89	125,07	126,21
L4	91,69	134,63	138,47
	64 nós		
L1	81,55	215,80	217,23
L2	85,94	214,89	215,12
L3	80,92	216,27	217,58
L4	79,71	217,07	217,98

Nos experimentos com concorrência, o aumento no número de réplicas foi benéfico. A explicação está no fato de que nestes experimentos ocorre

desbalanceamento de carga e o balanceamento dinâmico ajuda a atenuar o problema, melhorando o desempenho geral.

Tempo de processamento de consultas isoladas: BME *versus* ParGRES

Para permitir uma visão geral no desempenho do processamento de consultas OLAP utilizando o ParGRES, realizamos uma análise comparativa entre os tempos das consultas (`Q1` a `Q14`) executadas no ambiente dos experimentos desta dissertação e os tempos dessas mesmas consultas executadas no servidor do projeto BME. Vale salientar que estamos comparando tempos de processamento de consultas executadas em ambientes computacionais bem diferentes, e essa comparação tem como objetivo mostrar em que ocasiões o ParGRES atingiu o desempenho do servidor BME, que possui configurações superiores. Na Tabela 20, apresentamos as características detalhadas do servidor BME.

Tabela 20: Características detalhadas do servidor BME

Servidor BME	
Modelo	IBM x3650 E5310
CPU	2x Intel Quad-Core Xeon 1.6 GHz
Memória	8 GB
Disco	6 x 146GB = 876GB
Sistema operacional	LINUX Red Hat Enterprise
SGBD	Banco de dados paralelo

As consultas Q1 a Q14 foram executadas diretamente no servidor BME apenas uma vez e foram utilizadas as mesmas consultas modificadas dos experimentos (e não as consultas geradas pela aplicação BME). Todos os tempos de processamento (em segundo) estão organizados na Tabela 21.

Com apenas 1 nó do agrupamento de computadores Paraquad, os tempos do servidor BME não foram superados; os tempos obtidos com o agrupamento foram muito maiores. A consulta `Q10`, demorou 35 vezes mais para ser processada e a `Q11`, demorou 70 vezes mais. `Q1` demorou apenas 2 vezes mais tempo, no entanto, foi a consulta que apresentou menor redução de tempo à medida em que nós eram adicionados à configuração do agrupamento. Com 2 nós do agrupamento Paraquad, 50% das consultas foram processadas em menos tempo do que no servidor BME; as consultas `Q7`, `Q8`, `Q10`, `Q11` e `Q12`, que demoraram mais tempo do que no BME, possuem

mais agregações que as demais; Q10, Q11 e Q12 realizam junções entre duas tabelas de fatos; e Q14 realiza junções entre três tabelas de fatos. Essas características envolvem alto custo de processamento. Com 4 e 8 nós, 75% das consultas foram processadas em menos tempo; as consultas Q7, Q8, Q11 e Q12 continuam com tempos superiores. As características comuns entre elas é o grande número de agregações e junções entre tabelas de fatos; embora Q7 e Q8 não realizem junções entre tabelas de fatos, elas realizam junções com a maior tabela de dimensão da base, que possui cerca de 5.500 tuplas. Com 32 e 64 nós, os tempos de todas as consultas são bem menores que o tempo do servidor BME. O processamento paralelo intra-consulta aliado ao número de nós disponíveis para o processamento contribuiu para a redução significativa dos tempos.

Tabela 21: Tempo de processamento de consultas (em segundos) BME x ParGRES

		Agrupamento de computadores Paraquad						
	BME	1 nó	2 nós	4 nós	8 nós	16 nós	32 nós	64 nós
Q1	18,72	31,90	13,59	9,00	7,86	2,61	2,50	2,36
Q2	171,38	217,41	113,30	24,41	17,85	13,70	7,18	2,50
Q3	168,07	217,47	107,60	16,81	13,96	7,94	5,78	2,19
Q4	174,73	217,90	101,76	16,89	12,88	12,36	4,79	2,24
Q5	141,39	217,44	88,77	10,70	10,38	9,12	4,47	1,69
Q6	137,38	217,62	92,48	11,45	11,42	7,44	1,91	1,75
Q7	3,70	217,76	85,98	12,33	10,55	5,60	1,51	1,35
Q8	5,72	217,67	86,19	10,08	9,01	8,35	2,30	1,81
Q9	265,74	460,30	208,19	41,30	22,52	11,03	5,71	3,60
Q10	17,38	586,65	244,96	16,18	10,86	8,86	5,70	1,70
Q11	8,04	568,54	273,66	13,35	12,74	11,54	2,30	1,82
Q12	134,75	1.242,22	618,85	174,95	84,45	42,25	21,91	11,43
Q13	173,06	218,37	112,56	23,15	11,68	8,15	5,83	2,94
Q14	231,71	1.030,08	511,11	93,80	48,08	25,50	14,45	9,20

6 Conclusão

O ParGRES é uma solução de código aberto, desenvolvida para ser uma camada intermediária entre um banco de dados e uma aplicação cliente em um agrupamento de banco de dados, provendo paralelismo inter e intra-consulta no processamento de consultas OLAP. Através de experimentos utilizando o benchmark TPC-H [41], o ParGRES apresentou alto desempenho durante o processamento de consultas, nos motivando a avaliar seu desempenho no processamento de consultas OLAP sobre uma base de dados real.

Os experimentos desta dissertação foram realizados utilizando a base de dados do Censo Demográfico 2000 – CD2000, pesquisa produzida pelo Instituto Brasileiro de Geografia e Estatística – IBGE, e quatorze consultas OLAP do Banco Multidimensional de Estatísticas – BME, um armazém de dados desenvolvido por esta instituição. A base de dados CD2000 é composta por três tabelas de fatos, totalizando aproximadamente 30 milhões de tuplas; e as consultas oriundas do BME possuem diferentes níveis de complexidade, incluindo diversas junções entre tabelas de fatos e de dimensões, muitas agregações e predicados de seleção com variados fatores de seletividade. Esta base de dados e as consultas OLAP formam um cenário típico de Sistemas de Informações de Apoio à Decisão, com grande volume de dados e consultas de análise complexa e de alto custo. O foco principal dos experimentos é analisar a abordagem não intrusiva da Fragmentação Virtual Adaptativa no paralelismo intra-consulta e o balanceamento de carga durante o processamento de consultas.

Foram conduzidos experimentos em basicamente dois cenários: com replicação total e com replicação parcial da base de dados nos nós do agrupamento de computadores. Esses experimentos foram divididos em duas etapas: em uma, as consultas foram executadas isoladamente com diferentes números de nós do agrupamento, e na outra, as consultas foram organizadas em lotes, executados concorrentemente, também com diferentes números de nós. O número de nós variou entre 1 e 64. Excelentes resultados foram obtidos no que tange a redução do tempo de processamento das consultas em ambos os cenários.

Nos experimentos com base totalmente replicada, os resultados obtidos foram muito bons, tanto nos experimentos com consultas isoladas quanto nos experimentos com lotes concorrentes, porque todas as consultas e lotes apresentaram redução no tempo de processamento à medida que se aumentou o número de nós. Com exceção de

apenas uma consulta, as consultas obtiveram aceleração super-linear em todas as configurações do agrupamento. Com configurações de 2 e 4 nós, ocorre uma redução significativa no tempo de processamento de todas as consultas, sendo as maiores acelerações super-lineares obtidas. O que explica essa aceleração é o fato de todos os fragmentos virtuais já poderem ser carregados na memória principal dos nós. Com configurações superiores, como 8, 16, 32 e 64 nós, a redução no tempo de processamento é menor, mas ainda assim obtêm-se aceleração super-linear em todos os casos. A consulta mais demorada leva cerca de 20 minutos para ser processada. Com 8 nós, esse tempo diminui para 1,41 minutos e com 64 nós, o tempo desta consulta diminui para apenas 11 segundos.

O tempo médio de processamento de um lote de consultas é de 1 hora e 44 minutos. Com configuração de 4 nós, ocorre uma redução significativa no tempo de processamento de todos os lotes, para aproximadamente 7,5 minutos. Com 64 nós, um lote pode ser processado em apenas 41,7 segundos. Comparando os tempos de execução dos lotes com concorrência e seus respectivos tempos quando executados isoladamente, observamos que os tempos obtidos com concorrência foram quase sempre muito bons. Mesmo levando mais tempo para serem processados em um ambiente com concorrência, percebemos ganhos nos tempos de processamento dos lotes à medida que novos nós foram adicionados. Assim, podemos concluir que o ParGRES teve um bom comportamento nesse cenário de concorrência em sistemas OLAP, mantendo o desempenho no processamento de consultas paralelas já observado no processamento de consultas sem concorrência.

Para avaliar o comportamento do ParGRES em bases parcialmente replicadas, foram feitas algumas modificações em seu código-fonte seguindo a abordagem do protótipo SmaQSS, que une a Fragmentação Virtual Adaptativa e a fragmentação física híbrida de banco de dados para prover paralelismo intra-consulta em bases de dados parcialmente replicadas. Essas modificações incluem o algoritmo de difusão de mensagens de oferta de ajuda e o projeto de distribuição do ParGRES. Tanto a difusão de mensagens como o projeto de distribuição têm como base o conceito de espalhamento encadeado.

Nos experimentos com bases parcialmente replicadas, observaram-se as mesmas reduções nos tempos de processamento à medida que aumentamos os números de nós, com exceção das configurações maiores que 32 nós, em que houve um ligeiro aumento no tempo de processamento das consultas. Para identificar a causa do comportamento

observado, realizamos experimentos nos quais coletamos o tempo total de processamento e o tempo de processamento individual de cada consulta em cada nó do agrupamento, identificando assim o custo envolvido na composição dos resultados das consultas. O custo de composição é o tempo gasto pelo ParGRES para organizar os resultados das consultas processadas em cada nó e enviá-lo para o usuário. Esses experimentos foram realizados com 32, 40, 48, 56 e 64 nós.

Com 32 nós, em bases total e parcialmente replicadas, a diferença entre o tempo médio total e individual é muito pequena, mantendo o tempo total muito próximo do tempo individual. Com 40 nós, na maioria das vezes já observamos aumento em ambos os tempos, total e individual, comparando esse tempo com a configuração anterior; a diferença entre os tempos mantém-se estável. Com 48 nós, a diferença entre os tempos atinge 42%, mas em grande parte, a diferença é baixa, em torno de 10%. Com 56 e 64 nós, o tempo médio total tem um aumento significativo sobre o tempo individual, fazendo com que o tempo de processamento seja maior do que o obtido com a configuração anterior, e essa diferença atinja até 374%. Em configurações de 64 nós detectamos as maiores diferenças entre o tempo total e o tempo individual. O custo envolvido na composição de resultados em configurações de agrupamentos maiores que 32 nós se tornou relevante para o tamanho dos experimentos desta dissertação, influenciando no tempo total do processamento das consultas.

Em todos os experimentos realizados foram coletadas as mensagens, de oferta e aceite de ajuda, enviadas pelos nós. No ParGRES, o balanceamento de carga dinâmico é feito através de ajuda entre os nós, ou seja, quando um nó acaba o processamento de seu trabalho ele pode ajudar outro nó. É importante salientar que um nó pode aceitar a ajuda oferecida, mas não ser ajudado. Portanto, a ajuda efetivamente dada é aquela em que o nó aceitou a ajuda e foi de fato ajudado por outro nó.

Observamos que em bases parcialmente replicadas com 1 e 2 réplicas, houve redução do número de mensagens trocadas em relação ao número em bases totalmente replicadas. Com 2 réplicas, o número de mensagens enviadas é maior que o número de mensagens enviadas com 1 réplica; e com base totalmente replicada, o número de mensagens é maior que com 2 réplicas. Esse comportamento nos leva a concluir que à medida que adicionamos réplicas, seguindo na direção da replicação total, aumenta o número de mensagens de oferta de ajuda e diminui a proporção de ajuda efetivamente dada. Além disso, os tempos de processamento das consultas em bases totalmente replicadas são maiores que os tempos com 2 réplicas, que são maiores que os tempos

com 1 réplica. Nesses experimentos detectamos que o desbalanceamento de carga foi muito pequeno.

Já nos experimentos com concorrência, o aumento no número de réplicas foi benéfico. Com base totalmente replicada, os tempos de processamento foram menores que com bases parciais, com 1 e 2 réplicas. À medida que o número de réplicas diminuiu, o tempo de processamento aumentou. A explicação está no fato de que nestes experimentos ocorre maior desbalanceamento de carga e o balanceamento dinâmico ajuda a atenuar o problema, melhorando o desempenho geral.

Analisando os tempos de processamento e a troca de mensagens de oferta e aceite de ajuda conjuntamente, podemos concluir que o envio de mensagens para um grupo maior de nós provoca uma degradação no desempenho de processamento, principalmente quando não existe desbalanceamento de carga (ou este é pequeno), pois há o custo envolvido em receber a ajuda, verificar se ela é necessária e em responder (aceitando ou não) a oferta. Logo, quanto maior o número de réplicas, maior o número de mensagens enviadas e maior o tempo de processamento, principalmente em cenários de baixo desbalanceamento de carga.

Os resultados obtidos durante os experimentos mostram que o ParGRES pode ser uma solução alternativa, com baixo custo de implementação em um agrupamento de computadores, para aumento de desempenho no processamento de consultas OLAP em cenários reais, tanto com bases totalmente replicadas quanto com bases parcialmente replicadas.

A utilização do ParGRES para a obtenção de paralelismo no processamento de consultas não acarreta qualquer alteração no projeto físico de uma base de dados durante a migração de um ambiente centralizado para um distribuído. No entanto, pode-se tirar proveito do projeto físico existente e aumentar ainda mais o desempenho do processamento de consultas. Para a realização deste processo de migração de banco de dados, propomos uma metodologia que fornece um conjunto de recomendações a serem seguidas para um melhor aproveitamento do uso do ParGRES na paralelização de consultas. O uso desta metodologia consiste na realização de duas etapas principais: de análise do esquema conceitual e do projeto físico do banco de dados a ser migrado; e de fragmentação e distribuição do Banco de Dados para o ParGRES.

As principais contribuições desta dissertação são:

1. Avaliação do desempenho do ParGRES no processamento de consultas OLAP em uma base de dados real.

2. Alterações no algoritmo de difusão de mensagens de oferta de ajuda e no projeto de distribuição do ParGRES, viabilizando o seu funcionamento em bases parcialmente replicadas bem como avaliação do seu desempenho nesse cenário.

3. Definição de uma metodologia para transformar uma aplicação baseada em consultas sequenciais OLAP em uma aplicação que possibilite a realização de consultas paralelas utilizando o ParGRES.

4. Viabilização de uma solução alternativa com baixo custo de implementação para aumento de desempenho no processamento de consultas OLAP em cenários reais.

Como desdobramento dos resultados obtidos, faz-se necessário um estudo sobre o algoritmo de composição de resultados, buscando uma forma de aperfeiçoá-lo para reduzir o seu tempo de execução. A composição dos resultados poderia ser feito em uma etapa paralela à geração dos resultados parciais, à medida que os resultados parciais fossem gerados e não como uma etapa extra, no final da geração de todos os resultados parciais.

Foi observado que o grande número de mensagens trocadas afeta o desempenho geral do processamento de consultas. Mais testes poderiam ser feitos, por exemplo, aumentando o número de mensagens desnecessárias enviadas pelos nós, para avaliar o quanto o desempenho geral é afetado por essa troca de mensagens.

O uso de replicação em banco de dados para aplicações OLAP não é viável no que tange o custo de armazenamento, porque esse tipo de banco possui grande volume de dados. Além disso, os resultados obtidos em nossos experimentos demonstraram que quanto maior a replicação menor o benefício no processamento das consultas quando não há desbalanceamento de carga. O funcionamento do ParGRES com bases parcialmente replicadas é de suma importância, pois torna mais atraente a sua adoção como solução para aumento de desempenho em aplicações OLAP. Nesta dissertação, nós fizemos as alterações no ParGRES neste sentido, mas é preciso que esse funcionamento seja mais amigável e acessível ao usuário final.

Outras pesquisas podem se beneficiar da base de dados e consultas reais utilizadas nesta dissertação para avaliação de seu desempenho em BDD, por exemplo, a realização de experimentos semelhantes ao desta dissertação em ambientes de Grade, utilizando o GParGRES, versão do ParGRES para Grade, proposta por KOTOWSKI [19].

Como desdobramento desta dissertação, destacamos um trabalho publicado (PAES [28]) na conferência científica "2008 *High Performance Computing for Computational Science*", o prêmio de melhor artigo "*Best Student Paper Award*" nesta

mesma conferência por este trabalho, e um trabalho a ser publicado na revista científica da série "*Lectures Notes in Computer Science*".

Referências Bibliográficas

1. AKAL, F., BÖHM, K., SCHEK, H-J. "OLAP Query Evaluation in a Database Cluster: a Performance Study on Intra-Query Parallelism". In: *Proceedings of the East European Conference on Advances in Databases and Information Systems (ADBIS), 6th European East Conference*, pp. 218-231, Bratislava, Slovakia, Setembro. 2002.
2. BARU, C. K., FECTEAU, G., GOYAL, A., HSIAO, H., JHINGRAN, A., PADMANABHAN, S., COPELAND, G. P., WILSON, W. G. "DB2 Parallel Edition". IBM System Journal, 1995.
3. BME, "Banco Multidimensional de Estatísticas". Disponível em: <http://www.bme.ibge.gov.br>. Acesso em: 10 mar. 2008.
4. CD2000, "Censo 2000". Disponível em: <http://www.ibge.gov.br/censo>. Acesso em: 10 mar. 2008.
5. CECCHET, E., MARGUERITE, J., PELTIER, M., MODRZYK, N., HANSEN, D., CARVALHO, N. "Sequoia User's Guide", 2006.
6. CECCHET, E., MARGUERITE, J., ZWAENEPOEL, W. "C-JDBC: Flexible Database Clustering Middleware". In: *Proceedings of USENIX Annual Technical Conference, Freenix Track*, pp.9-18. Boston, EUA, Junho. 2004.
7. CODD, E. F., CODD, S. B., SALLEY, C. T. "Providing OLAP (On-Line Analytical Processing) to Users-Analysts: An IT Mandate". Arbor Software, Technical Report, 1993.
8. CROWL, L., 1994, "How to Measure, Present and Compare Parallel Performance", *IEEE Parallel and Distributed Technology*, v.2, n.1, pp. 9-25.
9. ELMASRI, R., NAVATHE, S.B. *Sistemas de Banco de Dados*. 4 ed. São Paulo, Pearson Addison Wesley, 2005.
10. FURTADO, C. S. *Fragmentação física e virtual de dados em um agrupamento de banco de dados*, Dissertação de Msc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.
11. GRID5000, "Grid5000 Project web site". Disponível em: <http://www.grid5000.fr>. Acesso em: 10 mar. 2008.
12. HONG, W., STONEBRAKER, M. "Optimization of Parallel Query Execution Plans in XPRS". In: *Proceedings of the First International Conference on Parallel and Distributed Information Systems (PDIS)*, pp. 218-225. Florida, EUA, Dezembro. 1991.
13. HSIAO H., DEWITT, D. J. "Chained Declustering: A New Availability Strategy for Multi-processor Database Machines". In: *Proceedings of the Sixth International Conference on Data Engineering (ICDE)*, pp. 456-465, Los Angeles, Califórnia, Fevereiro. 1990.
14. HSQL Database Engine. Disponível em: <http://hsqldb.org/>. Acesso em: 10 mar. 2008.
15. IBGE, "Censo Demográfico 2000: Características Gerais da População – Resultados da Amostra". IBGE, Rio de Janeiro, 2003.
16. IBGE, "Instituto Brasileiro de Geografia e Estatística". Disponível em: <http://www.ibge.gov.br>. Acesso em: 10 mar. 2008.

17. KABRA, N., DEWITT, D. J. "Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, v. 1, pp. 106-117. ACM Press, New York. 1998.

18. KIMBALL, R., ROSS, M., MERZ, R. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2 ed. New York, John Wiley & Sons, 2002.

19. KOTOWSKI, N., LIMA, A. A., PACITTI, E., VALDURIEZ, P., MATTOSO, M. L. Q.: Parallel Query Processing for OLAP in Grids. Concurrency and Computation. Practice & Experience, http://dx.doi.org/10.1002/cpe.1303 (2008)

20. LEI nº 5.534, "Dispõe sobre a obrigatoriedade das informações estatísticas e dá outras providências.". Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/L5534.htm>. Acesso em: 10 mar. 2008.

21. LIMA, A. A. B. *Paralelismo Intra-Consulta em Clusters de Banco de Dados*. Tese de Dsc., COPPE/UFRJ, Rio de Janeiro, RJ, Brazil, 2004.

22. LIMA, A. A. B., MATTOSO, M., VALDURIEZ, P. "Adaptive Virtual Partitioning for OLAP Query Processing in a Database Cluster". In: *Proceedings of the 19th Brazilian Symposium on Database Systems (SBBD)*, pp.92-105. Brasília, Brazil, Outubro. 2004.

23. MACHADO, F. *Projeto de Data Warehouse: Uma Visão Multidimensional*. São Paulo, Érica, 2000.

24. MATTOSO, M., ZIMBRÃO, G., LIMA, A. A. B., BAIÃO. F., BRAGANHOLO, V., AVELEDA, A., MIRANDA, B., ALMENTERO, B., COSTA, M.N. "ParGRES Middleware for Executing OLAP Queries in Parallel". In: http://pargres.nacad.ufrj.br/Documentos/ES-690.pdf , Technical report, 2005.

25. MIRANDA, B., LIMA, A. A. B., VALDURIEZ, P., MATTOSO, M. "Apuama: Combining Intra-query and Inter-query Parallelism in a Database Cluster". In: *Currents Trends in Database Technology (EDBT)*, LNCS, v. 4254, pp.649-661. Springer, Heidelberg, 2006.

26. O'NEIL, P., QUASS, D. "Improved Query Performance with Variant Indexes". In: *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 38–49, Maio. 1997.

27. ÖSZU, T., VALDURIEZ, P. *Principles of Distributed Database Systems*. 2 ed. New Jersey, Prentice-Hall, 1999.

28. PAES, M., LIMA, A. A., VALDURIEZ, P., MATTOSO, M. L. Q. "High-performance Query Processing of a Real-world OLAP Database with ParGRES". In: *VECPAR'08 8th International Meeting High Performance Computing for Computational Science*, Toulousse, França, 2008.

29. ParGRES. Disponível em: <http://pargres.nacad.ufrj.br>. Acesso em: 10 mar. 2008.

30. PostgreSQL, "PostgreSQL Brasil". Disponível em: <http://postgresql.org.br>. Acesso em: 10 mar. 2008.

31. POWERDB, "ETH Zürich The Database Research Group". Disponível em: <http://www.dbs.ethz.ch/archive/index.html>. Acesso em: 10 mar. 2008.

32. RÖHM, U., BOHM, K. SCHEK, H-J. "Cache-Aware Query Routing in a Cluster of Databases". In: *Proceedings of the 17th International Conference on Data Engineering (ICDE)*, pp.641-650. IEEE Computer Society, Heidelberg, Germany, Abril. 2001.

33. RÖHM, U., BÖHM, K., SCHECK, H.-J., SCHULDT, H. "FAS - A Freshness-Sensitive Coordination Middleware for a Cluster of OLAP Components". In: *Proceedings of the 28th International Conference on Very Large Databases Conference (VLDB)*, pp.754-765. Hong Kong, China, Agosto. 2002.

34. RÖHM, U., BOHM, K., SCHEK, H-J. "OLAP Query Routing and Physical Design in a Database Cluster". In: *Proceedings of the 7th International Conference on Extending Database Technology: Advances in Database Technology (EDBT)*, pp.254-268. Springer-Verlag, London, UK, Março. 2000.

35. Sequoia Project. Disponível em: <http://sequoia.continuent.org/HomePage>. Acesso em: 10 mar. 2008.

36. STERLING, T. "An Introduction to PC Clusters for High Performance Computing". *The International Journal of High Performance Computing Applications* v.15, n.2, pp. 92-101. 2001.

37. TANDEM Database Group. "Nonstop SQL, a distributed high-performance, high-reliability implementation of SQL". In: *Workshop on High Performance Transaction Systems*, Asilomar, CA, 1987.

38. TERADATA. "DBC/1012 Database Computer System Manual Release 2.0". Terada Corporation, Documento Técnico Nº C10-0001-02, 1985.

39. THOMSEN, E. OLAP *Construindo Sistemas de Informações Multidimensionais*. 2 ed. Rio de Janeiro, Campus, 2002.

40. TPC Benchmark C. Disponível em: <http://www.tpc.org/tpcc>. Acesso em: 10 mar. 2008.

41. TPC Benchmark H. Disponível em: <http://www.tpc.org/tpch/>. Acesso em: 10 mar. 2008.

42. TPC Benchmark W. Disponível em: <http://www.tpc.org/tpcw/default.asp>. 10 mar. 2008.

43. VALDURIEZ, P. "Parallel Database Systems: Open Problems and New Issues". *International Journal on Distributed and Parallel Databases* v.1, n.2, pp. 137-165. 1993.

44. WOJCIECHOWSKA, I. Broadcasting in Grid Graphs. Tese de Dsc., West Virginia University, College of Engineering and Mineral Resources, West Virginia, EUA, 1999.

Anexo A – Tabelas de fatos

Neste anexo são apresentados os atributos das tabelas de fatos `CD00AMDOMI`, `CD00AMFAMI` e `CD00AMPESS` da base de dados AmCD2000 utilizada nos experimentos. Todos os atributos descritos abaixo são do tipo numérico.

Tabela CD00AMDOMI

ATRIBUTOS (variáveis)	DESCRIÇÃO DOS ATRIBUTOS
CODANOPESQ	Ano da pesquisa
CODPAIS	País
CODUFCENSO	Unidade da Federação (uf)
CODMESO	Mesorregião
CODMICRO	Microrregião
CODMUNICIPIO	Município
CODDISTRITO	Distrito
CODSUBDIST	Sub-distrito
CONTROLE	Controle
V0400	Domicílio
CODREGGEOGR	Região geográfica
CODREGMETRO	Região metropolitana
CODV1005	Situação do setor
CODV1006	Situação do domicílio
CODV1007	Tipo do setor
V0110	Total de homens
CODV0110	Total de homens, classe
V0111	Total de mulheres
CODV0111	Total de mulheres, classe
CODV0201	Espécie
CODV0202	Tipo do domicílio
V0203	Total de cômodos
CODV0203	Total de cômodos, classe
V0204	Total de cômodos servindo de dormitório
CODV0204	Total de cômodos servindo de dormitório, classe
CODV0205	Condição de ocupação do domicílio
CODV0206	Condição de ocupação do terreno
CODV0207	Origem de abastecimento de água
CODV0208	Tipo de canalização do abastecimento de água
CODV0209	Total de banheiros
CODV0210	Existência de sanitário
CODV0211	Forma de esgotamento sanitário
CODV0212	Destino do lixo
CODV0213	Existência de iluminação elétrica
CODV0214	Existência de rádio
CODV0215	Existência de geladeira ou freezer
CODV0216	Existência de videocassete
CODV0217	Existência de máquina de lavar roupa
CODV0218	Existência de forno de microondas
CODV0219	Existência de linha telefônica instalada
CODV0220	Existência de microcomputador
CODV0221	Número de televisores
CODV0222	Número de automóveis para uso particular
CODV0223	Número de aparelhos de ar condicionado
V7100	Total de moradores no domicílio

CODV7100	Total de moradores no domicilio, classe
V7203	Densidade de moradores por cômodo
CODV7203	Densidade de moradores por cômodo, classe
V7204	Densidade de moradores por dormitório
CODV7204	Densidade de moradores por dormitório, classe
V7401	Número de componentes da família 01
V7402	Número de componentes da família 02
V7403	Número de componentes da família 03
V7404	Número de componentes da família 04
V7405	Número de componentes da família 05
V7406	Número de componentes da família 06
V7407	Número de componentes da família 07
V7408	Número de componentes da família 08
V7409	Número de componentes da família 09
V7616	Total de rendimentos do domicílio particular
V7617	Total de rendimentos do domicílio particular, em salários mínimos
PESODOMI	Peso do domicílio
CODV1111	Existência de identificação
CODV1112	Existência de iluminação pública
CODV1113	Existência de calçamento/pavimentação

Tabela CD00AMPESS

ATRIBUTOS (variáveis)	DESCRIÇÃO DOS ATRIBUTOS
CODANOPESQ	Ano da pesquisa
CODPAIS	País
CODUFCENSO	Uf
CODMESO	Mesorregião
CODMICRO	Microrregião
CODMUNICIPIO	Município
CODDISTRITO	Distrito
CODSUBDIST	Subdistrito
CONTROLE	Controle
V0400	> 00 (pessoa)
CODREGMETRO	Região metropolitana
CODAREAP	Área de ponderação
CODREGGEOGR	Região geográfica
CODV0401	Sexo
CODV0402	Relação com responsável pelo domicílio
CODV0403	Relação com responsável pela família
V0404	Número da família
V4752	Idade calculada em anos completos - a partir de 1 ano
CODV4752	Idade em anos, classe
V4754	Idade calculada em meses – menos de um ano (valores de 00 a 11)
CODV0408	Cor ou raça
CODV4090	Código da religião
CODV0410	Problema mental permanente
CODV0411	Capacidade de enxergar
CODV0412	Capacidade de ouvir
CODV0413	Capacidade de caminhar / subir escadas
CODV0414	Deficiências
CODV0415	Sempre morou neste município
V0416	Tempo de moradia neste município
CODV0417	Nasceu neste município

CODV0418	Nasceu nesta uf
CODV0419	Nacionalidade
V0420	Ano que fixou residência no brasil
CODV4210	Código da uf ou país de nascimento
V0422	Tempo de moradia na uf
CODV4230	Código da uf ou país de residência anterior
CODV0424	Residência em 31 de julho de 1995
CODV4250	Código do município de residência
CODV4260	Código da uf ou país de residência em 31/07/1995
CODV4276	Código do município e uf ou país estrangeiro que trabalha ou estuda
CODV0428	Sabe ler e escrever
CODV0429	Freqüenta escola ou creche
CODV0430	Curso que freqüenta
CODV0431	Série que freqüenta
CODV0432	Curso mais elevado que freqüentou, concluindo pelo menos uma série
CODV0433	Última série concluída com aprovação
CODV0434	Concluiu o curso no qual estudou
CODV4355	Código do curso mais elevado concluído
CODV4300	Anos de estudo
CODV0436	Vive em companhia de cônjuge ou companheiro(a)
CODV0437	Natureza da última união
CODV0438	Estado civil
CODV0439	Na semana de 23 a 29 de julho de 2000, trabalhou remunerado
CODV0440	Na semana, tinha trabalho mas estava afastado
CODV0441	Na semana, ajudou sem remuneração, no trabalho exercido por pessoa moradora do domicílio, ou como aprendiz / estagiário.
CODV0442	Na semana, ajudou sem remuneração, no trabalho exercido por pessoa moradora do domicílio em atividade. De cultivo, extração vegetal...
CODV0443	Na semana, trabalhou em cultivo, etc, para alimentação de pessoas moradoras no domicílio
CODV0444	Quantos trabalhos, tinha na semana de 23 a 29 de julho de 2000
CODV4452	Código novo da ocupação
CODV4462	Código novo da atividade
CODV0447	Nesse trabalho era...
CODV0448	Era empregado pelo rjfp ou como militar
CODV0449	Quantos empregados trabalhavam nessa firma
CODV0450	Era contribuinte de instituto de previdência oficial
CODV4511	Não tem rendimento no trabalho principal
V4512	Rendimento bruto no trabalho principal
V4513	Total de rendimentos no trabalho principal
V4514	Total de rendimentos no trabalho principal, em salários mínimos
CODV4521	Não tem rendimento nos demais trabalhos
V4522	Rendimento bruto nos demais trabalhos
V4523	Total de rendimentos nos demais trabalhos
V4524	Total de rendimentos nos demais trabalhos, em salários mínimos
V4525	Total de rendimentos em todos os trabalhos
V4526	Total de rendimentos em todos os trabalhos, em salários mínimos
V0453	Horas trabalhadas por semana no trabalho principal
V0454	Horas trabalhadas nos demais trabalhos
V4534	Total de horas trabalhadas
CODV0455	Providência p/ conseguir trabalho
CODV0456	Em julho de 2000, era aposentado de instituto de previdência oficial
V4573	Rendimento de aposentadoria, pensão
V4583	Rendimento de aluguel

V4593	Rendimento de pensão alimentícia, mesada, doação
V4603	Rendimento de renda mínima, bolsa-escola, seguro-desemprego
V4613	Outros rendimentos
V4614	Total de rendimentos
V4615	Total de rendimentos, em salários mínimos
V4620	Total de filhos nascidos vivos
V0463	Total de filhos nascidos vivos que estavam vivos
V4654	Idade calculada do último filho nascido vivo
V4670	Total de filhos nascidos mortos
V4690	Total de filhos tidos
CODV4690	Total de filhos tidos, classe
PESOPESS	Peso da pessoa
V4621	Filhos nascidos vivos: homens
V4622	Filhos nascidos vivos: mulheres
V4631	Filhos que estavam vivos: homens
V4632	Filhos que estavam vivos: mulheres
CODV0464	Sexo do último filho nascido vivo
V4671	Filhos nascidos mortos: homens
V4672	Filhos nascidos mortos: mulheres
CODV4354	Código do curso mais elevado concluído (concla)
CODV4219	Código da uf ou país de nascimento (onu)
CODV4239	Código da uf ou país (onu) de residência anterior
CODV4269	Código da uf ou país (onu) de residência em 31/07/1995
CODV4279	Código do país estrangeiro (onu) que trabalha ou estuda
CODV4451	Código antigo da ocupação
CODV4461	Código antigo da atividade

Tabela CD00AMFAMI

ATRIBUTOS (variáveis)	DESCRIÇÃO DOS ATRIBUTOS
CODANOPESQ	Ano da pesquisa
CODPAIS	Pais
CODUFCENSO	Unidade da Federação
CODMESO	Mesorregião
CODMICRO	Microrregião
CODMUNICIPIO	Município
CODDISTRITO	Distrito
CODSUBDIST	Sub-distrito
CONTROLE	Controle
V0404	> 00 (número de ordem da família no domicílio)
CODREGGEOGR	Região geográfica
CODREGMETRO	Região metropolitana
AREAP	Área de ponderação
CODV0404	Tipo de família (1)
CODV0404_2	Tipo de família (2)
V4614B	Rendimento nominal familiar
CODV4615B	Classe de rendimento nominal familiar
V4614C	Rendimento nominal, responsável/casal
CODV4615C	Classe de rendimento nominal, responsável/casal
V4616_7400	Rendimento nominal familiar per-capita
CODV4615_7400	Classe de rendimento nominal familiar per-capita
V7400	Número de componentes da família
CODV7400	Classe de número de componentes

V7400A	Número de componentes homens da família
CODV7400A	Classe de número de componentes homens
V7400B	Número de componentes mulheres da família
CODV7400B	Classe de número de componentes mulheres
PESOFAMI	Peso da família

Anexo B – Tabelas de dimensões

Neste anexo são apresentados os atributos e descrições das tabelas de dimensões da base de dados `AmCD2000` utilizada nos experimentos. Todas as dimensões possuem os mesmos atributos (`CODIGO` e `DENOMINACAO`), com exceção das dimensões espacial e temporal, que possuem um atributo a mais, `IND_EXIBICAO`.

Tabelas de dimensões – atributos

ATRIBUTOS (variáveis)	DESCRIÇÃO	TIPO
CODIGO	Código da classe	NUMÉRICO
DENOMINACAO	Descrição da classe	VARCHAR2
IND_EXIBICAO	Classe exibível ou não para o usuário	CHAR

Tabelas de dimensões espacial e temporal – descrição

DIMENSÃO	DESCRIÇÃO DA DIMENSÃO
T004	Ocorrência temporal do Censo Demográfico 2000
G000	País
G031	Regiões Geográficas (Grandes Regiões)
G032	Unidades da Federação
G033	Mesorregião
G034	Microrregião
G035	Municípios
G036	Distrito
G037	Subdistrito
G039	Região Metropolitana
G042	Área de ponderação

Dimensões que se relacionam com CD00AMFAMI – descrição

DIMENSÃO	DESCRIÇÃO DA DIMENSÃO
M290	Número de componentes, classe
M291	Número de componentes homens, classe
M292	Número de componentes mulheres, classe
M293	Família, ordem
M295	Família, tipo
M296	Rendimento nominal, responsável/casal, classe
M297	Rendimento nominal, familiar, classe

Dimensões que se relacionam com CD00AMDOMI – descrição

DIMENSÃO	DESCRIÇÃO DA DIMENSÃO
M003	Total de cômodos, classe
M075	Existência de sanitário, de iluminação elétrica, de rádio, de geladeira ou freezer, de videocassete, de máquina de lavar roupa, de linha de telefone instalada, de forno de

	microondas, de microcomputador
M078	Existência de identificação, de iluminação pública
M102	Tipo do domicílio
M103	Condição de ocupação do domicílio
M104	Condição de ocupação do terreno
M105	Origem do abastecimento de água
M106	Canalização do abastecimento de água
M109	Destino do lixo
M115	Situação do setor
M116	Tipo do setor
M128	Densidade de moradores por cômodo, classe
M129	Densidade de moradores por dormitório, classe
M208	Situação do domicílio
M209	Total de homens, classe
M233	Existência de calçamento/pavimentação
M270	Espécie do domicílio
M272	Total de cômodos servindo de dormitório, classe
M273	Número de banheiros
M274	Forma do esgotamento sanitário
M275	Número de televisores
M276	Número de automóveis para uso particular
M277	Número de aparelhos de ar condicionado
M278	Total de moradores no domicilio, classe

Dimensões que se relacionam com CD00AMPESS – descrição

DIMENSÃO	DESCRIÇÃO DA DIMENSÃO
M159	Raça ou cor
M167	Nacionalidade
M298	Pessoa, condição no domicílio
M300	Pessoa, sexo
M301	Pessoa, condição na família
M302	Pessoa, idade em anos, classe
M307	Religião ou culto
M308	Capacidade de caminhar
M309	Deficiência, tipo
M311	Natural do Município
M314	Residência em 31/07/1995
M315	Residência em 31/07/1995, Código do Município
M320	Pessoa, alfabetização
M321	Estudante, escola ou creche
M322	Estudante, curso
M323	Estudante, série
M324	Não estudante, curso mais elevado freqüentado
M325	Não estudante, última série concluída com aprovação
M326	Não estudante, curso concluído
M327	Não estudante, curso concluído , espécie
M330	Cônjuge, Existência
M331	Última união, Natureza
M332	Estado civil, tipo
M333	Trabalhos, número
M334	Trabalho principal, ocupação, código 2000
M338	Trabalho principal, setor de atividade, código 2000
M341	Trabalho principal, posição da ocupação
M342	Empregador, número de empregados

M343	Contribuinte previdenciário
M348	Filhos tido, total, classe
M350	Filho, último nascido vivo, sexo
M355	Sim / Não / Menor de 10 anos
M361	Deficiência, existência
M4210	Código da UF ou país de nascimento/residência
M4219	Código da uf ou país de nascimento/residência (ONU)
M4230	Código da UF ou país de residência anterior
M4239	Código da uf ou país de residência anterior (ONU)
M4276	Código da UF ou país estrangeiro que trabalha ou estuda
M4279	Código do país estrangeiro que trabalha ou estuda (ONU)
M4300	Anos de estudo
M4354	Código do curso mais elevado concluído (CONCLA)
M4511	Não tem rendimento

Anexo C – Chaves estrangeiras

Neste anexo são apresentados os atributos e tabelas das chaves estrangeiras das tabelas de fatos da base de dados `AmCD2000` utilizada nos experimentos.

Tabela CD00AMDOMI – chaves estrangeiras

	Referência em:	
CHAVE ESTRANGEIRA	TABELA	ATRIBUTO
CODANOPESQ	T004	CODIGO
CODPAIS	G000	CODIGO
CODREGGEOGR	G031	CODIGO
CODUFCENSO	G032	CODIGO
CODMESO	G033	CODIGO
CODMICRO	G034	CODIGO
CODMUNICIPIO	G035	CODIGO
CODDISTRITO	G036	CODIGO
CODSUBDISTR	G037	CODIGO
CODREGMETRO	G039	CODIGO
CODV0110	M209	CODIGO
CODV0201	M270	CODIGO
CODV0202	M102	CODIGO
CODV0203	M003	CODIGO
CODV0204	M272	CODIGO
CODV0205	M103	CODIGO
CODV0206	M104	CODIGO
CODV0207	M105	CODIGO
CODV0208	M106	CODIGO
CODV0209	M273	CODIGO
CODV0210	M075	CODIGO
CODV0211	M274	CODIGO
CODV0212	M109	CODIGO
CODV0213	M075	CODIGO
CODV0214	M075	CODIGO
CODV0215	M075	CODIGO
CODV0216	M075	CODIGO
CODV0217	M075	CODIGO
CODV0218	M075	CODIGO
CODV0219	M075	CODIGO

CODV0220	M075	CODIGO
CODV0221	M275	CODIGO
CODV0222	M276	CODIGO
CODV0223	M277	CODIGO
CODV1005	M115	CODIGO
CODV1006	M208	CODIGO
CODV1007	M116	CODIGO
CODV1111	M078	CODIGO
CODV1112	M078	CODIGO
CODV1113	M233	CODIGO
CODV7100	M278	CODIGO
CODV7203	M128	CODIGO
CODV7204	M129	CODIGO

Tabela CD00AMPESS – chaves estrangeiras

	Referência em:	
CHAVE ESTRANGEIRA	TABELA	ATRIBUTO
CODANOPESQ	T004	CODIGO
CODPAIS	G000	CODIGO
CODREGGEOGR	G031	CODIGO
CODUFCENSO	G032	CODIGO
CODMESO	G033	CODIGO
CODMICRO	G034	CODIGO
CODMUNICIPIO	G035	CODIGO
CODDISTRITO	G036	CODIGO
CODSUBDISTR	G037	CODIGO
CODREGMETRO	G039	CODIGO
CODV0401	M300	CODIGO
CODV0402	M298	CODIGO
CODV0403	M301	CODIGO
CODV0408	M159	CODIGO
CODV0410	M361	CODIGO
CODV0411	M308	CODIGO
CODV0412	M308	CODIGO
CODV0413	M308	CODIGO
CODV0414	M309	CODIGO
CODV0415	M311	CODIGO
CODV0417	M311	CODIGO
CODV0418	M311	CODIGO
CODV0419	M167	CODIGO
CODV0424	M314	CODIGO
CODV0428	M320	CODIGO
CODV0429	M321	CODIGO
CODV0430	M322	CODIGO
CODV0431	M323	CODIGO
CODV0432	M324	CODIGO
CODV0433	M325	CODIGO
CODV0434	M326	CODIGO
CODV0436	M330	CODIGO
CODV0437	M331	CODIGO
CODV0438	M332	CODIGO
CODV0439	M355	CODIGO

CODV0440	M355	CODIGO
CODV0441	M355	CODIGO
CODV0442	M355	CODIGO
CODV0443	M355	CODIGO
CODV0444	M333	CODIGO
CODV0447	M341	CODIGO
CODV0448	M355	CODIGO
CODV0449	M342	CODIGO
CODV0450	M343	CODIGO
CODV0455	M355	CODIGO
CODV0456	M355	CODIGO
CODV0464	M350	CODIGO
CODV4219	M4219	CODIGO
CODV4239	M4239	CODIGO
CODV4269	M4219	CODIGO
CODV4090	M307	CODIGO
CODV4210	M4210	CODIGO
CODV4230	M4230	CODIGO
CODV4250	M315	CODIGO
CODV4260	M4210	CODIGO
CODV4276	M4276	CODIGO
CODV4279	M4279	CODIGO
CODV4300	M4300	CODIGO
CODV4354	M4354	CODIGO
CODV4355	M327	CODIGO
CODV4452	M334	CODIGO
CODV4462	M338	CODIGO
CODV4511	M4511	CODIGO
CODV4521	M4511	CODIGO
CODV4690	M348	CODIGO
CODV4752	M302	CODIGO

Tabela CD00AMFAMI – chaves estrangeiras

	Referência em:	
CHAVE ESTRANGEIRA	**TABELA**	**ATRIBUTO**
CODANOPESQ	T004	CODIGO
CODPAIS	G000	CODIGO
CODREGGEOGR	G031	CODIGO
CODUFCENSO	G032	CODIGO
CODMESO	G033	CODIGO
CODMICRO	G034	CODIGO
CODMUNICIPIO	G035	CODIGO
CODDISTRITO	G036	CODIGO
CODSUBDISTR	G037	CODIGO
CODREGMETRO	G039	CODIGO
CODV04042	M295	CODIGO
CODV0404	M293	CODIGO
CODV4615_7400	M297	CODIGO
CODV4615B	M297	CODIGO
CODV4615C	M296	CODIGO

CODV7400	M290	CODIGO
CODV7400A	M291	CODIGO
CODV7400B	M292	CODIGO

Anexo D – Gráficos de resultados

Neste anexo são apresentados todos os gráficos dos tempos de processamento usados durante a análise dos resultados.

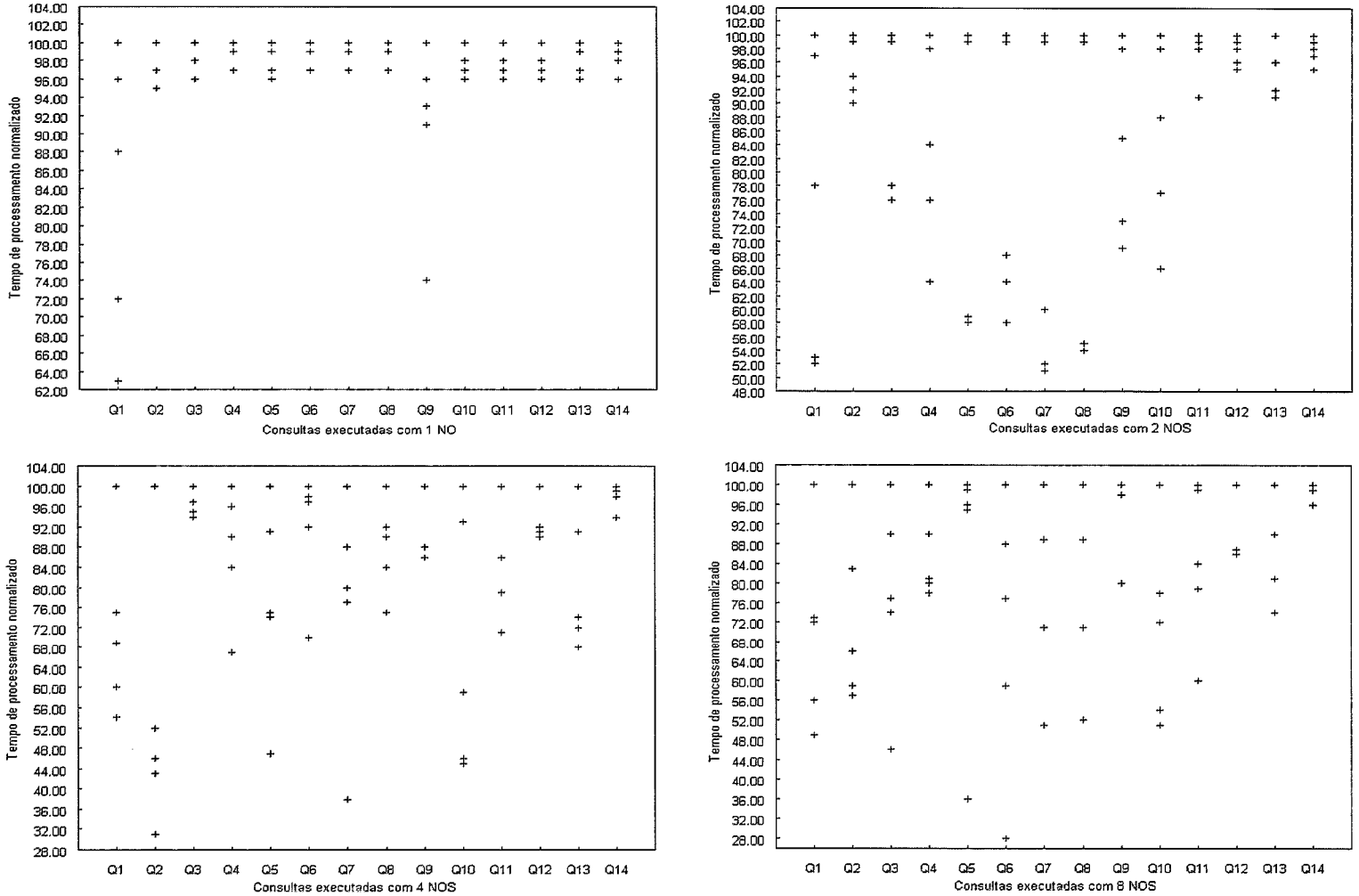


Figura 1: Tempos de execução de cada consulta por número de nós (1, 2, 4 e 8 nós) – BASE TOTALMENTE REPLICADA

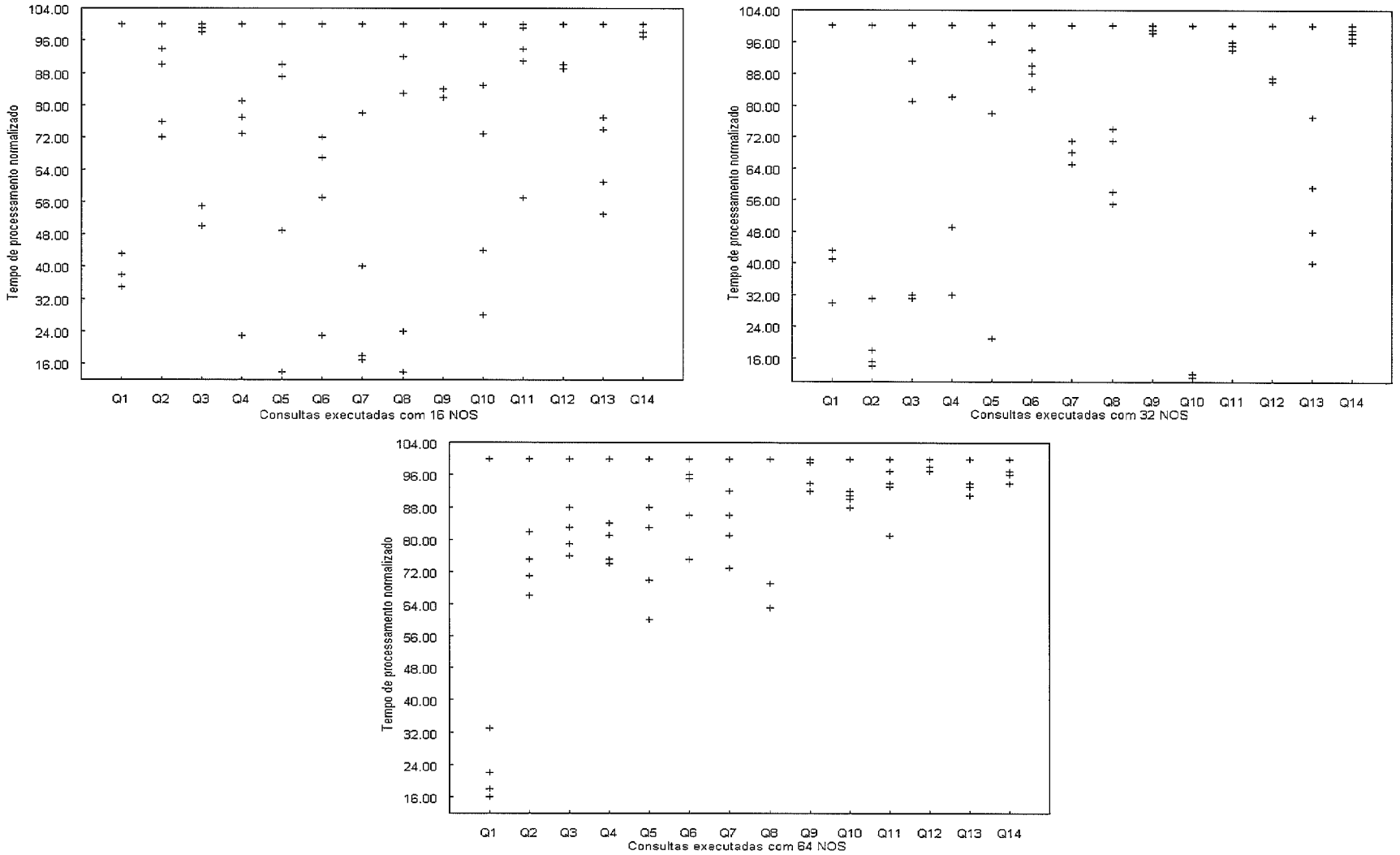


Figura 2: Tempos de execução de cada consulta por número de nós (16, 32 e 64 nós) - BASE TOTALMENTE REPLICADA

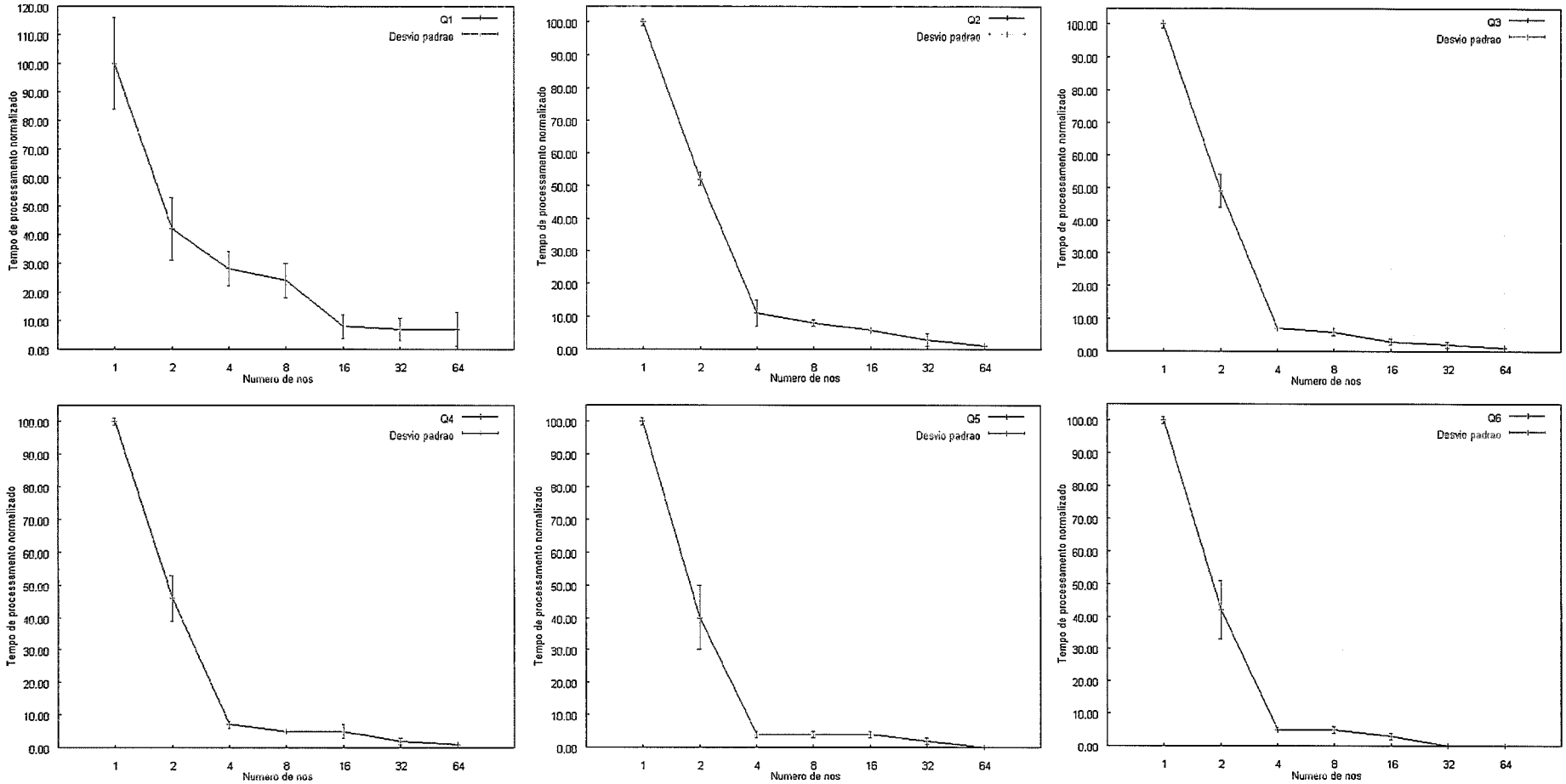


Figura 3: Desvio padrão das médias do tempo de execução - Consultas Q1 a Q6 - BASE TOTALMENTE REPLICADA

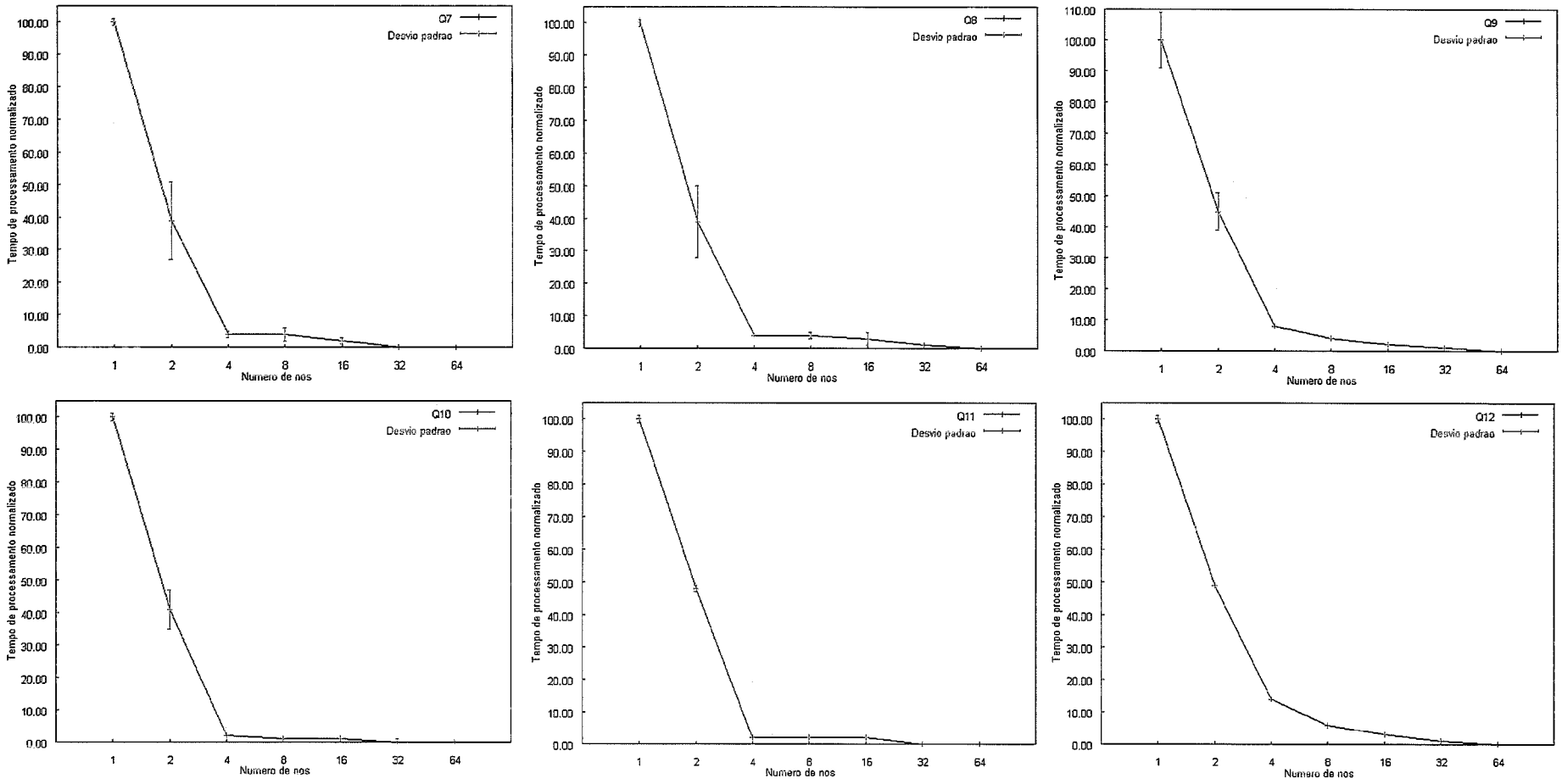
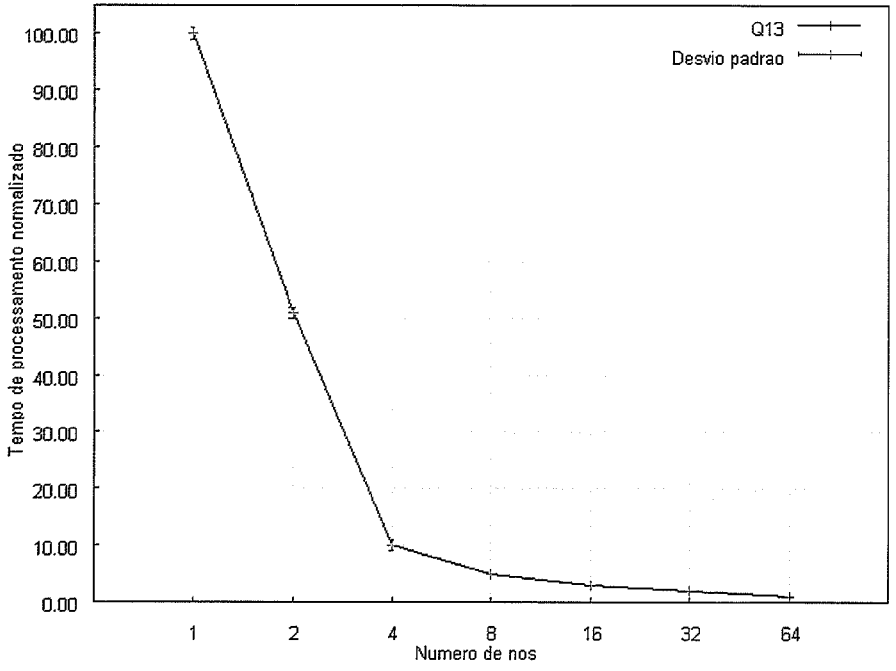


Figura 4: Desvio padrão das médias do tempo de execução - Consultas Q7 a Q12 - BASE TOTALMENTE REPLICADA



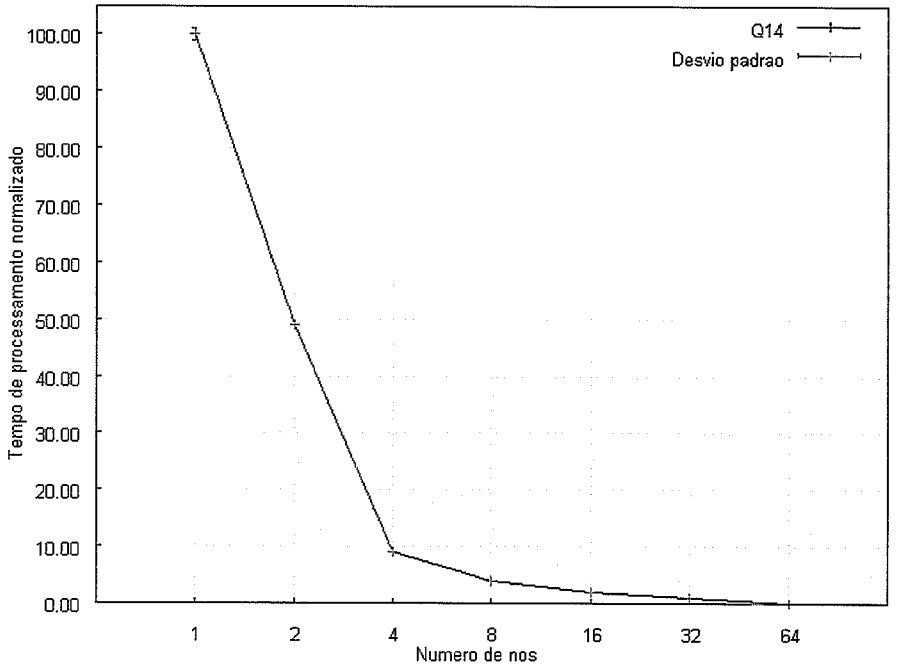


Figura 5: Desvio padrão das médias do tempo de execução - Consultas Q13 e Q14 - BASE TOTALMENTE REPLICADA

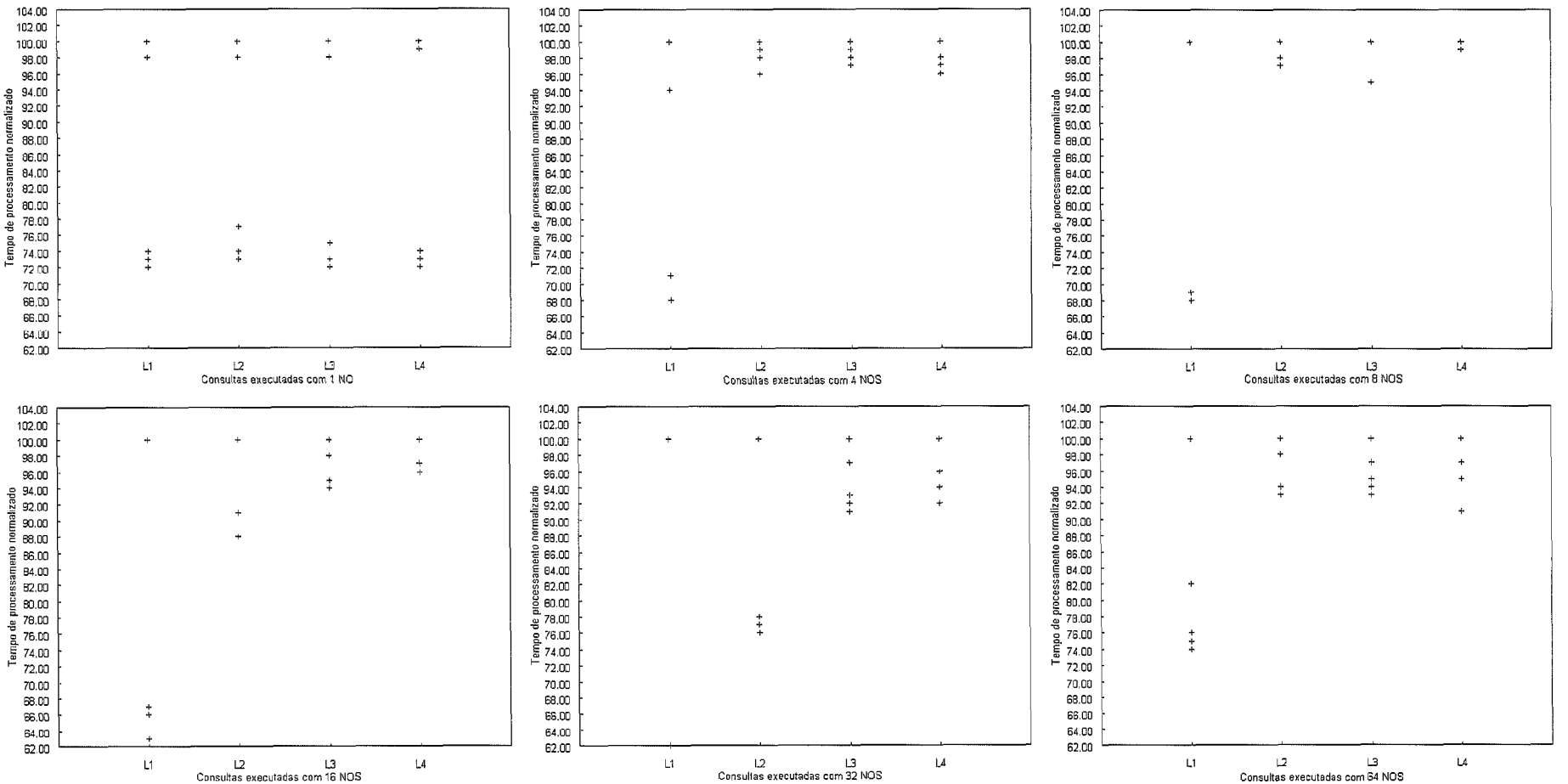


Figura 6: Tempos de execução de cada lote por número de nós - **Teste de força** - BASE TOTALMENTE REPLICADA

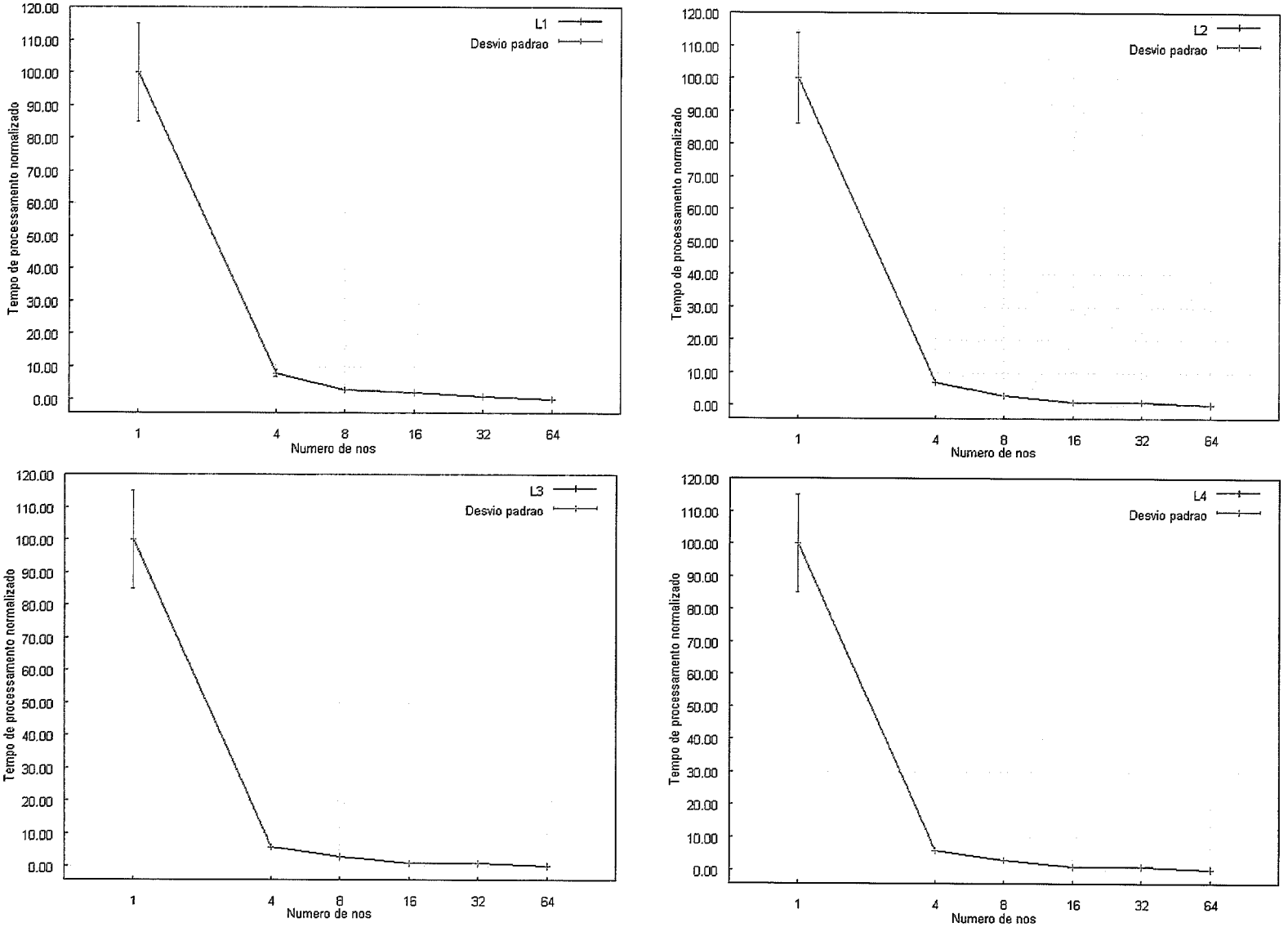


Figura 7: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de força** - BASE TOTALMENTE REPLICADA

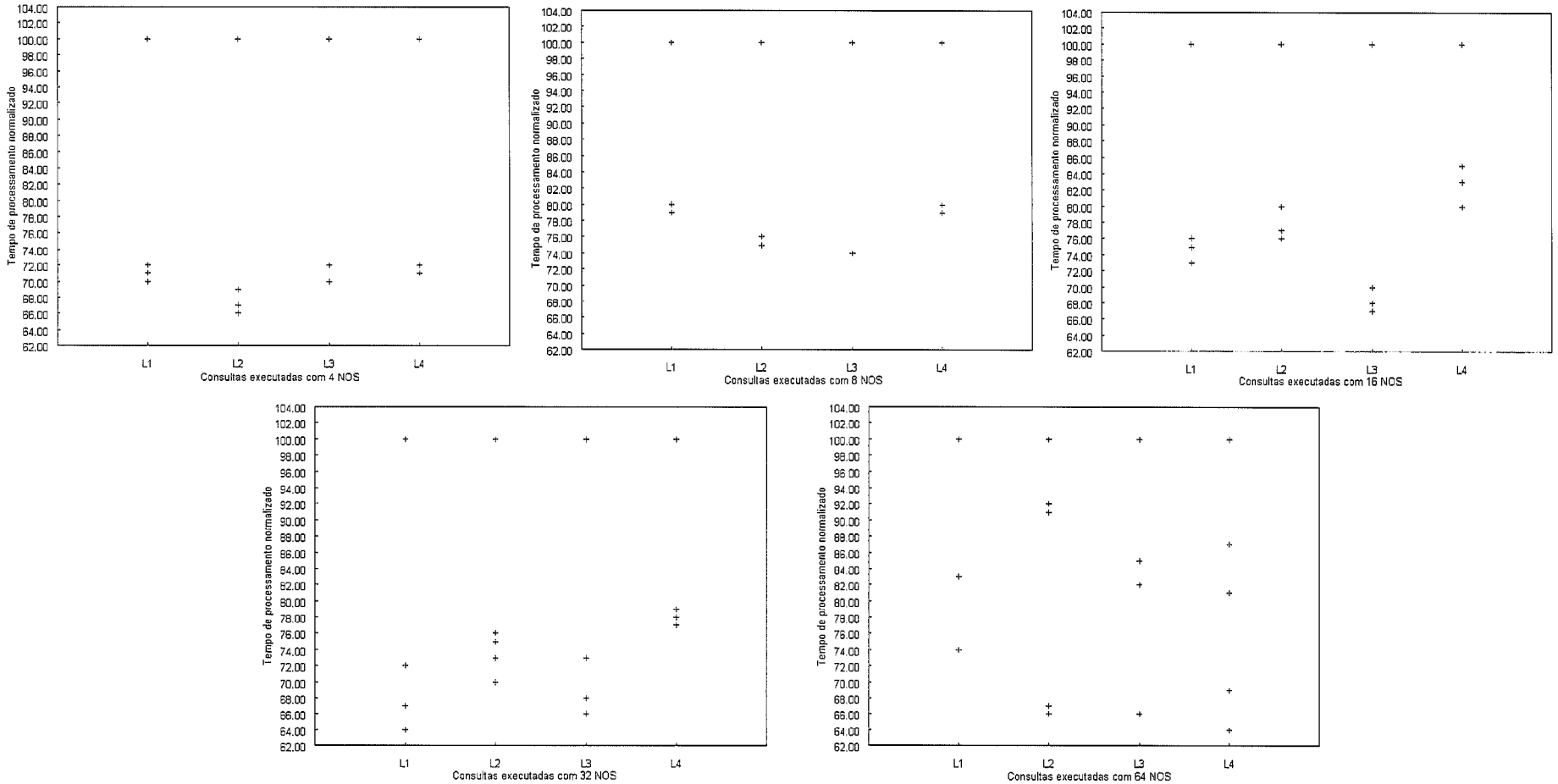


Figura 8: Tempos de execução de cada lote por número de nós - **Teste de carga** - BASE TOTALMENTE REPLICADA

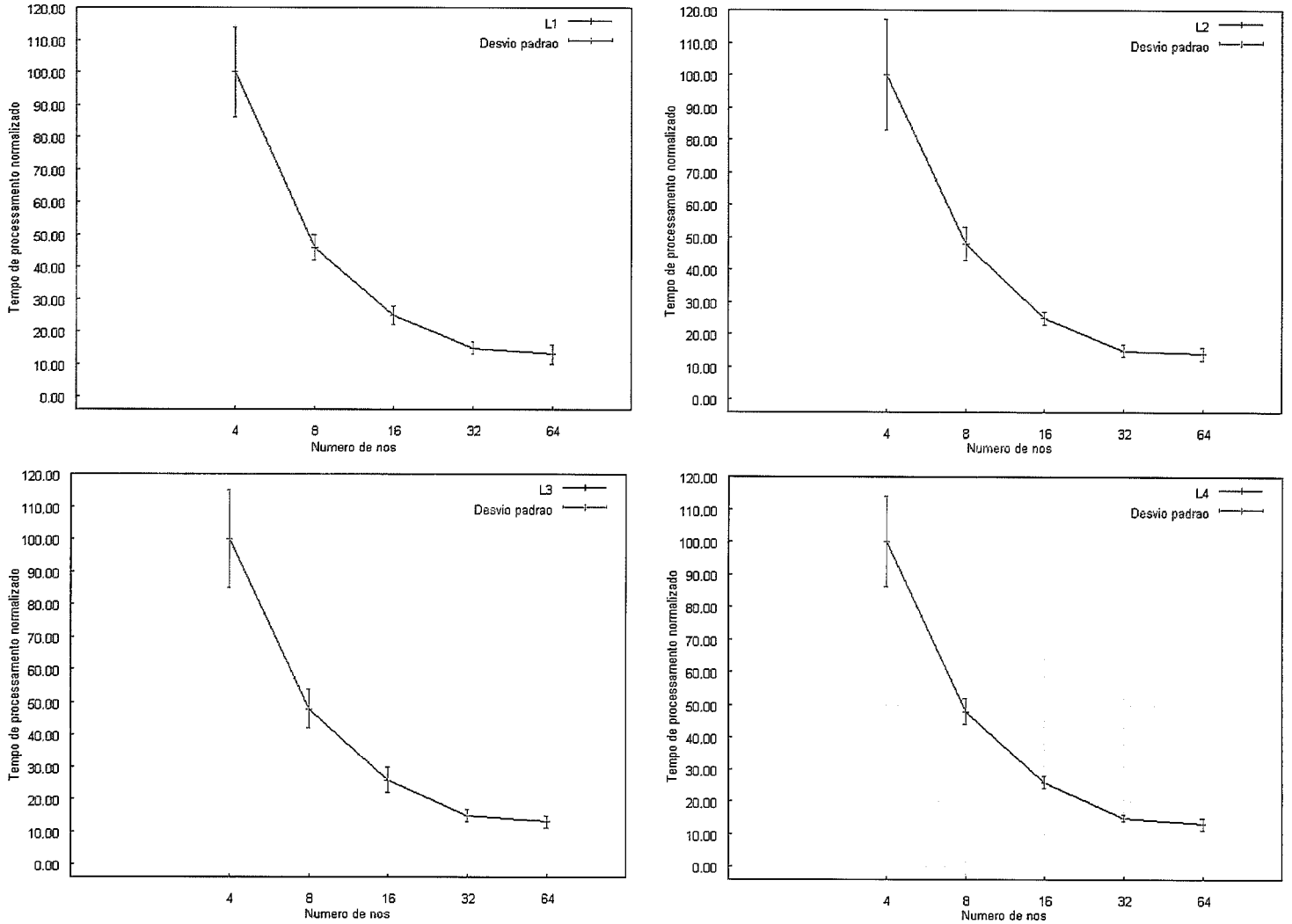


Figura 9: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de Carga** - BASE TOTALMENTE REPLICADA

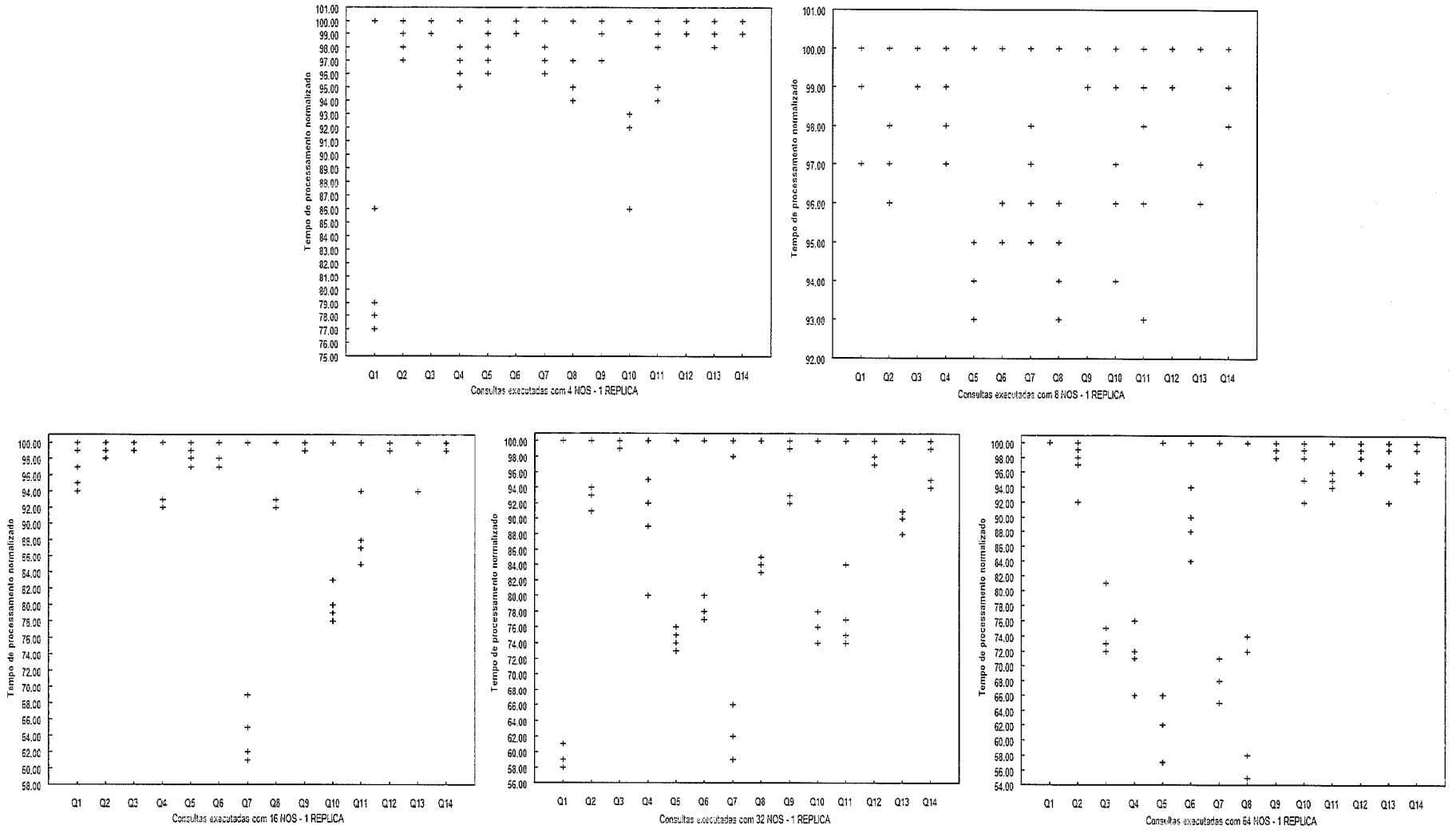


Figura 10: Tempos de execução de cada consulta por número de nós – BASE PARCIALMENTE REPLICADA – 1 RÉPLICA

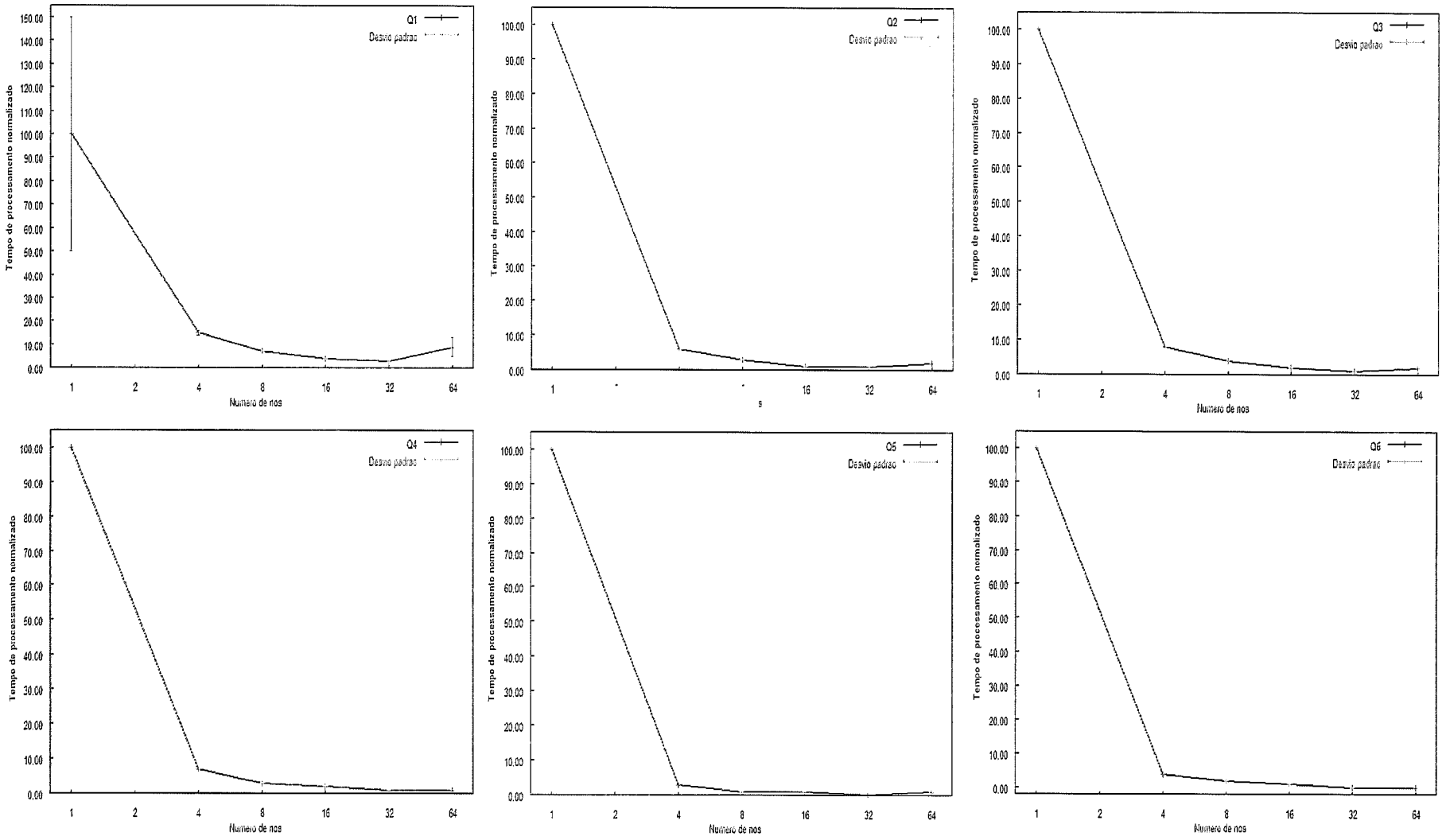


Figura 11: Desvio padrão das médias do tempo de execução - Consultas Q1 a Q6 - BASE PARCIALMENTE REPLICADA - 1 RÉPLICA

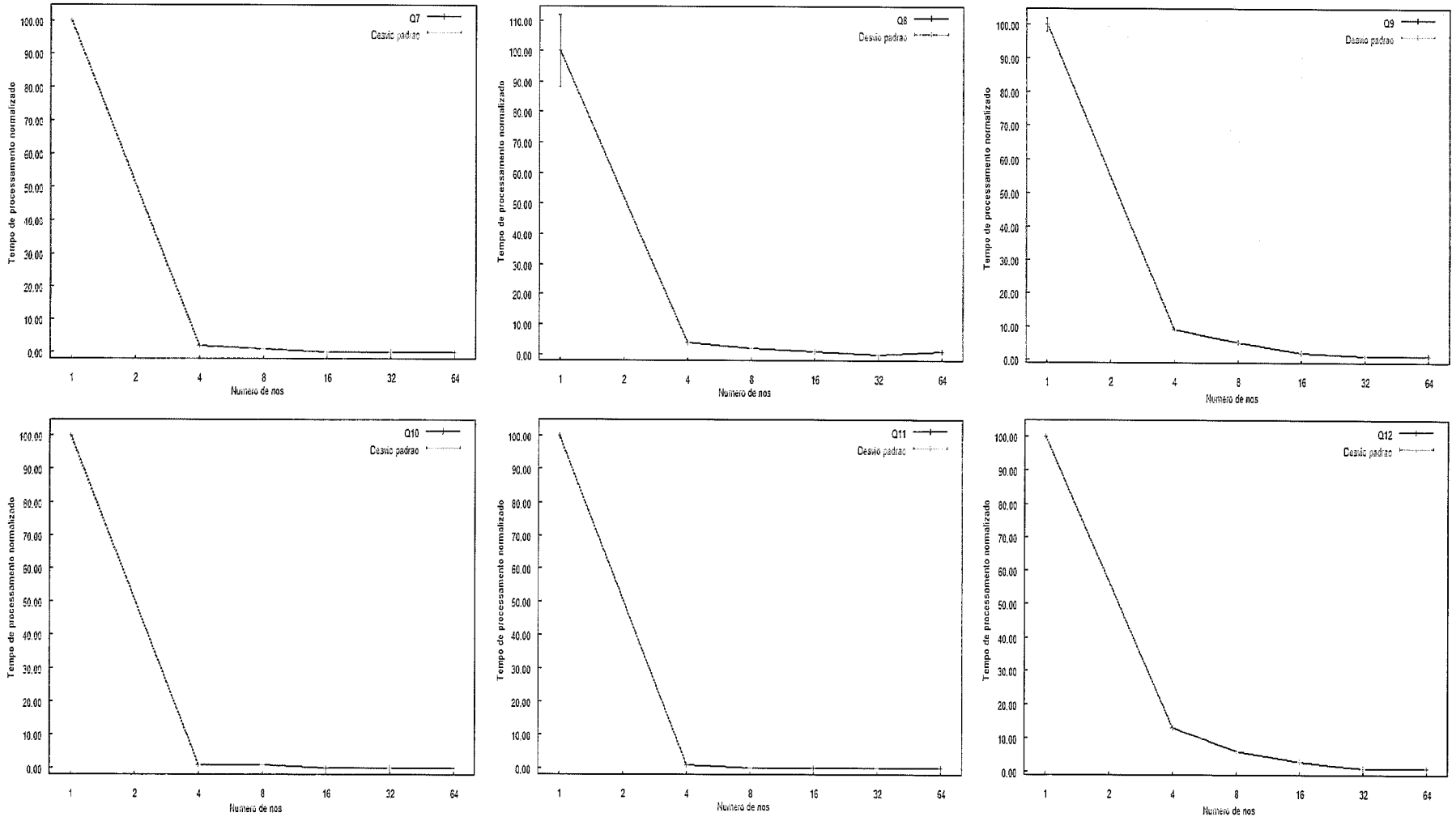
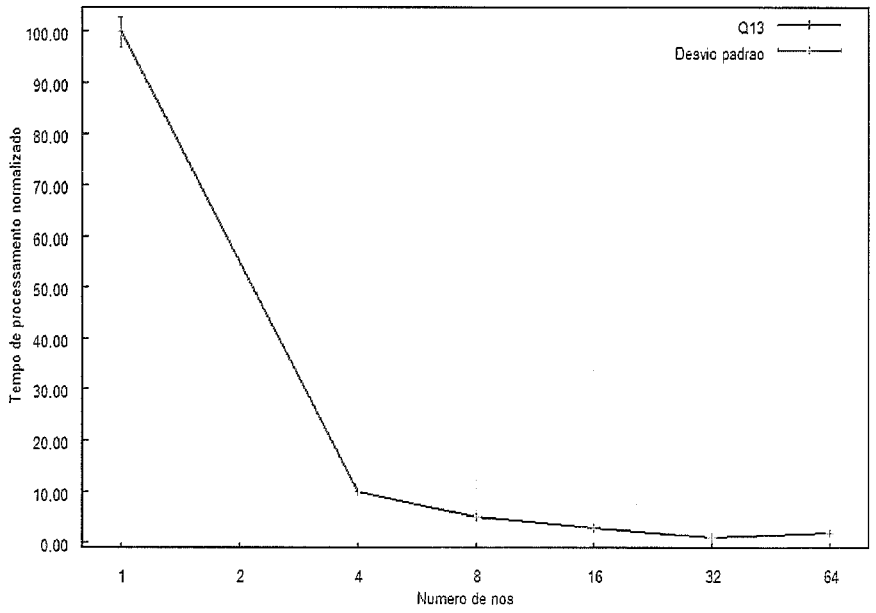


Figura 12: Desvio padrão das médias do tempo de execução - Consultas Q7 a Q12 - BASE PARCIALMENTE REPLICADA - 1 RÉPLICA



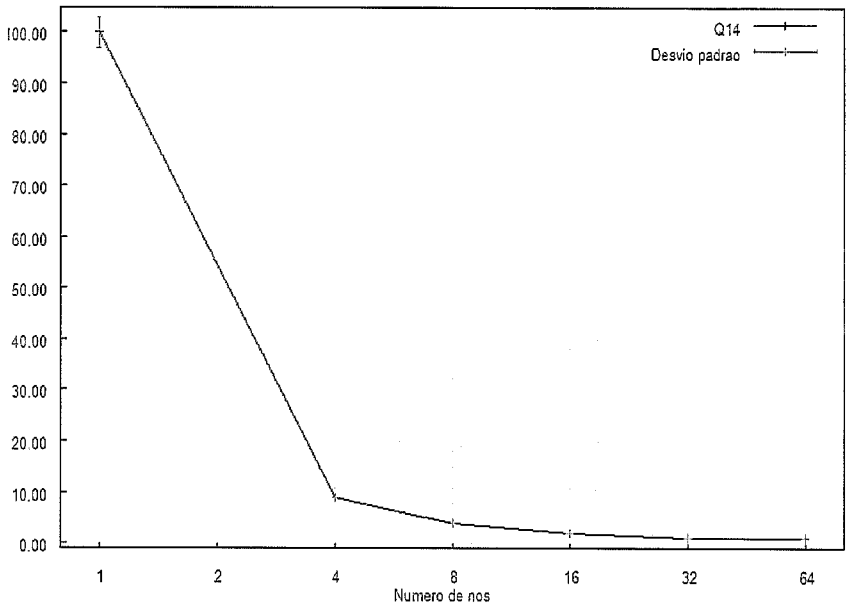


Figura 13: Desvio padrão das médias do tempo de execução - Consultas Q13 e Q14 - BASE PARCIALMENTE REPLICADA - 1 RÉPLICA

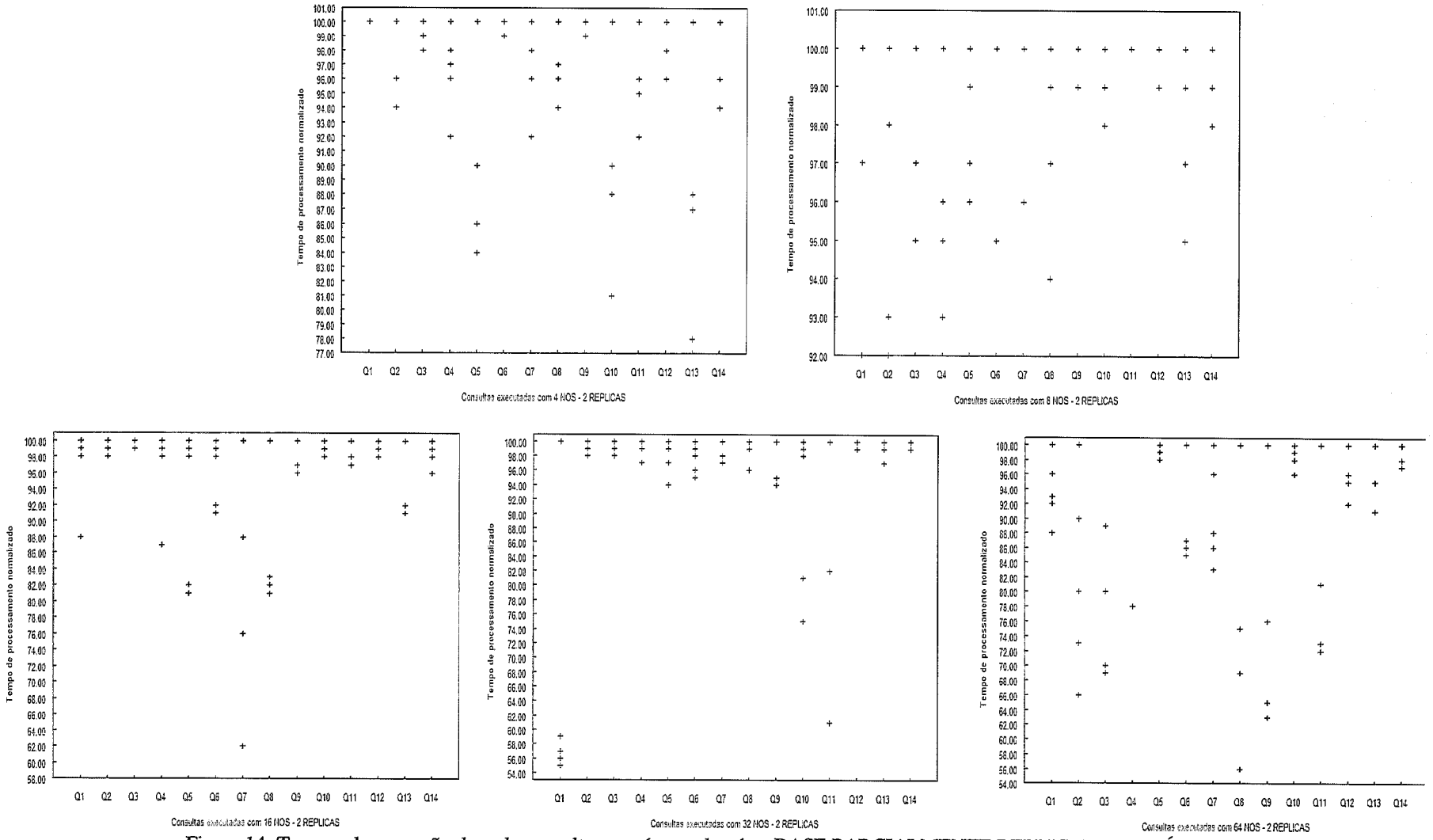


Figura 14: Tempos de execução de cada consulta por número de nós – BASE PARCIALMENTE REPLICADA – 2 RÉPLICAS

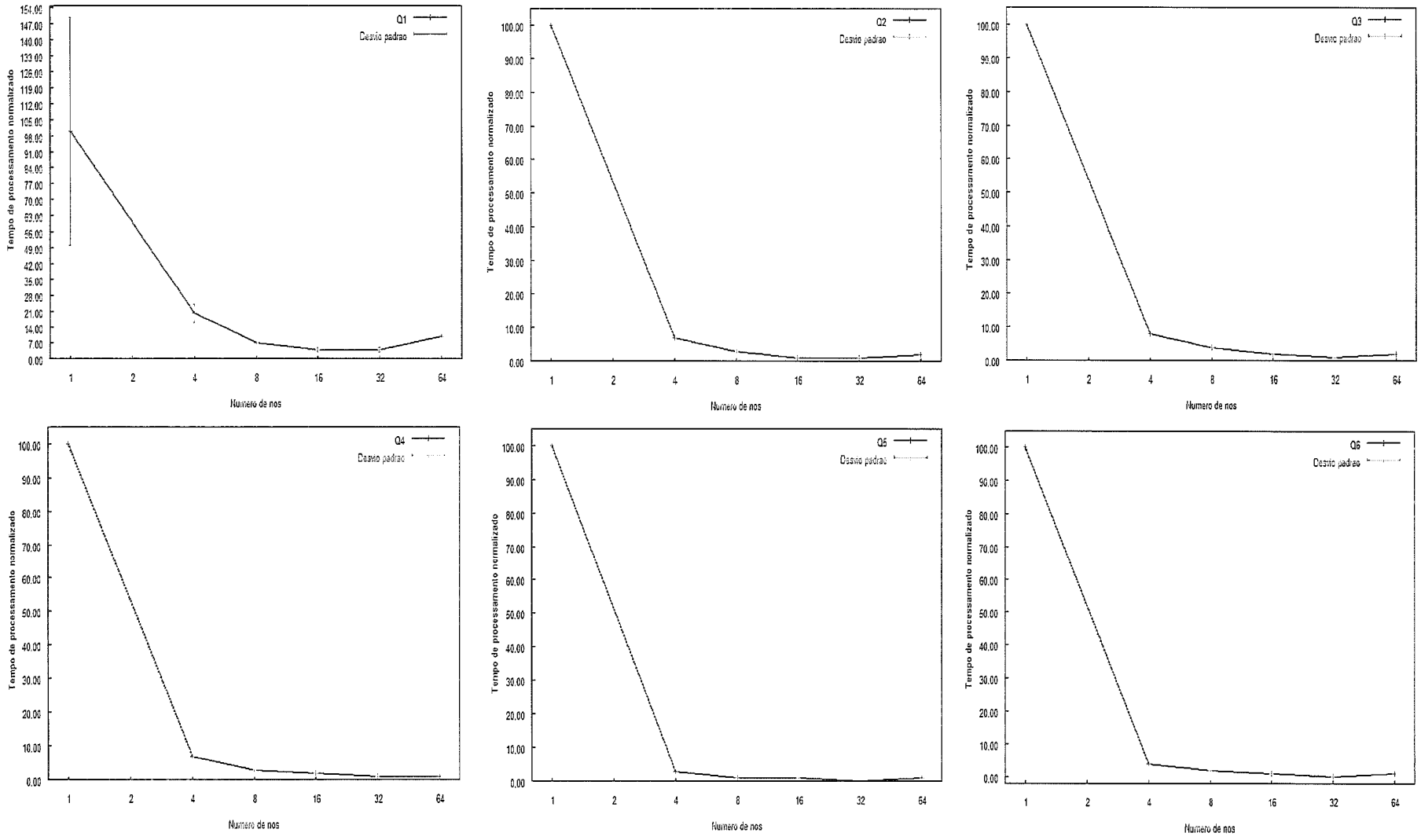


Figura 15: Desvio padrão das médias do tempo de execução - Consultas Q1 a Q6 - BASE PARCIALMENTE REPLICADA - 2 RÉPLICAS

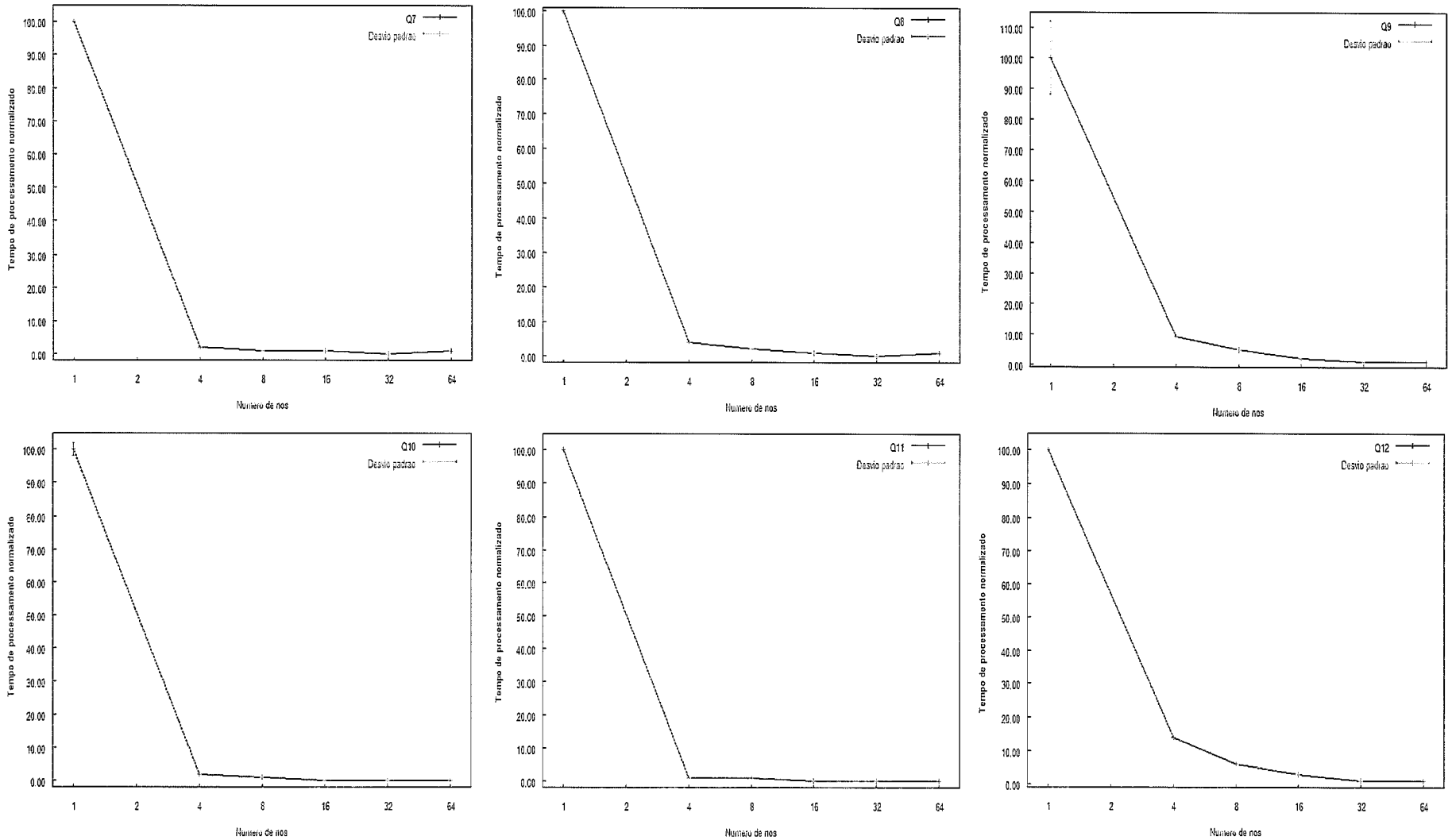


Figura 16: Desvio padrão das médias do tempo de execução - Consultas Q7 a Q12 - BASE PARCIALMENTE REPLICADA - 2 RÉPLICAS

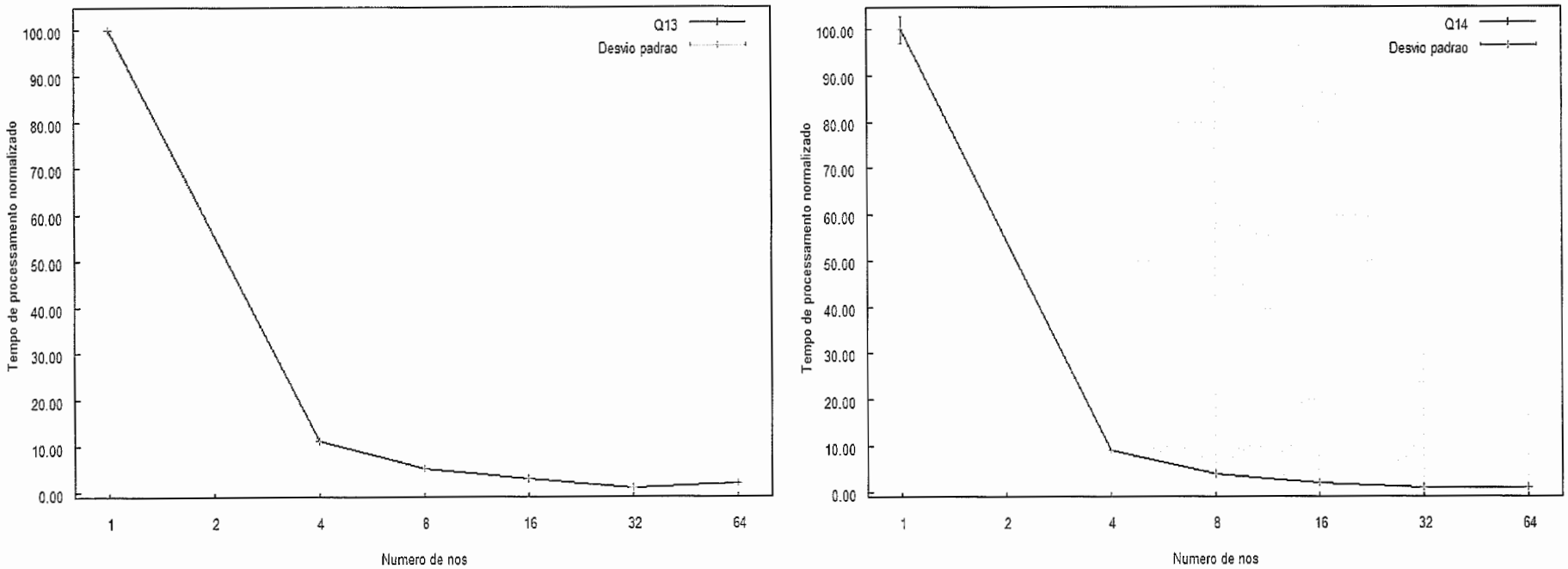


Figura 17: Desvio padrão das médias do tempo de execução - Consultas Q13 e Q14 - BASE PARCIALMENTE REPLICADA - 2 RÉPLICAS

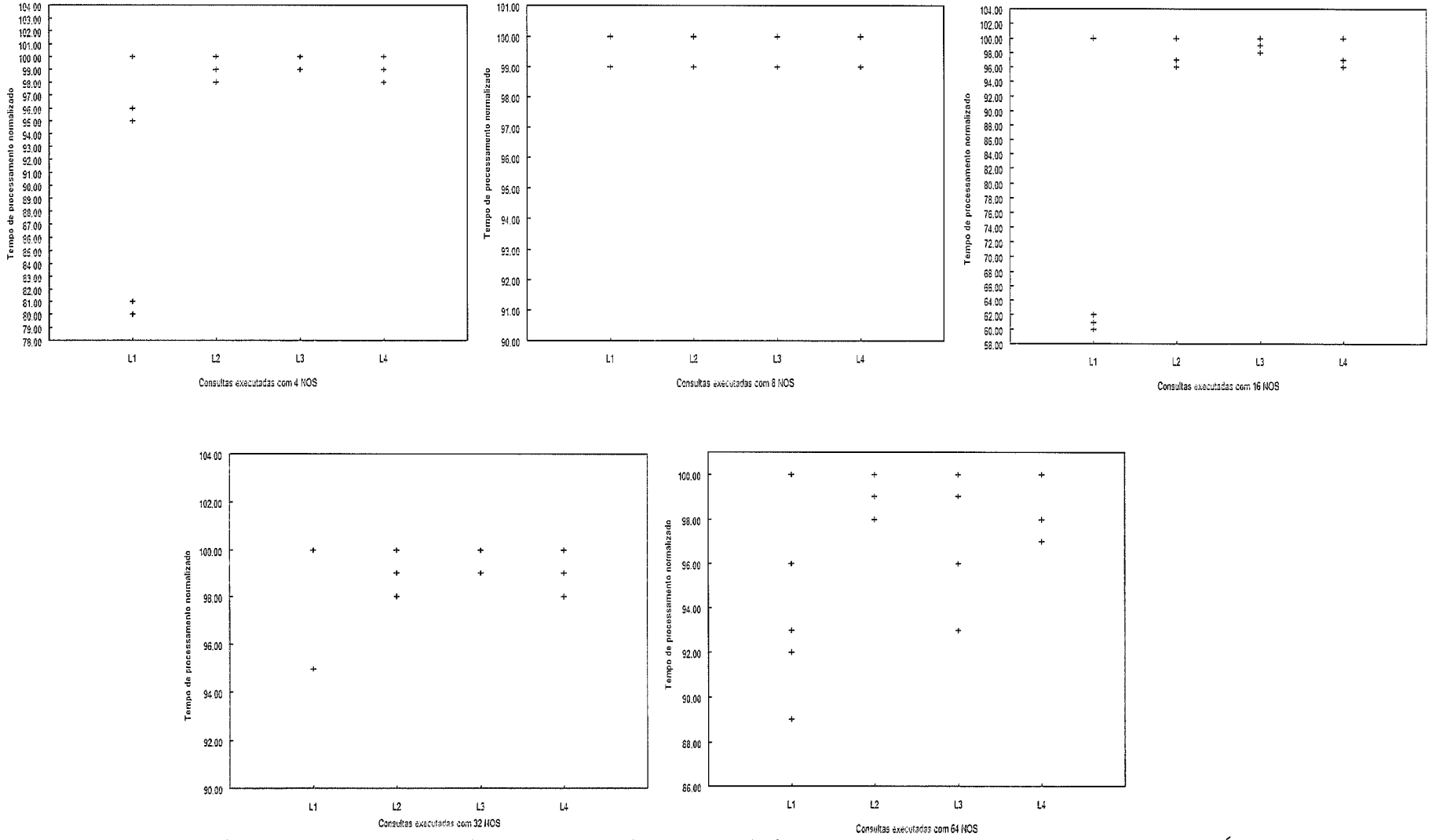


Figura 18: Tempos de execução de cada lote por número de nós - **Teste de força** - BASE PARCIALMENTE REPLICADA – 1 RÉPLICA

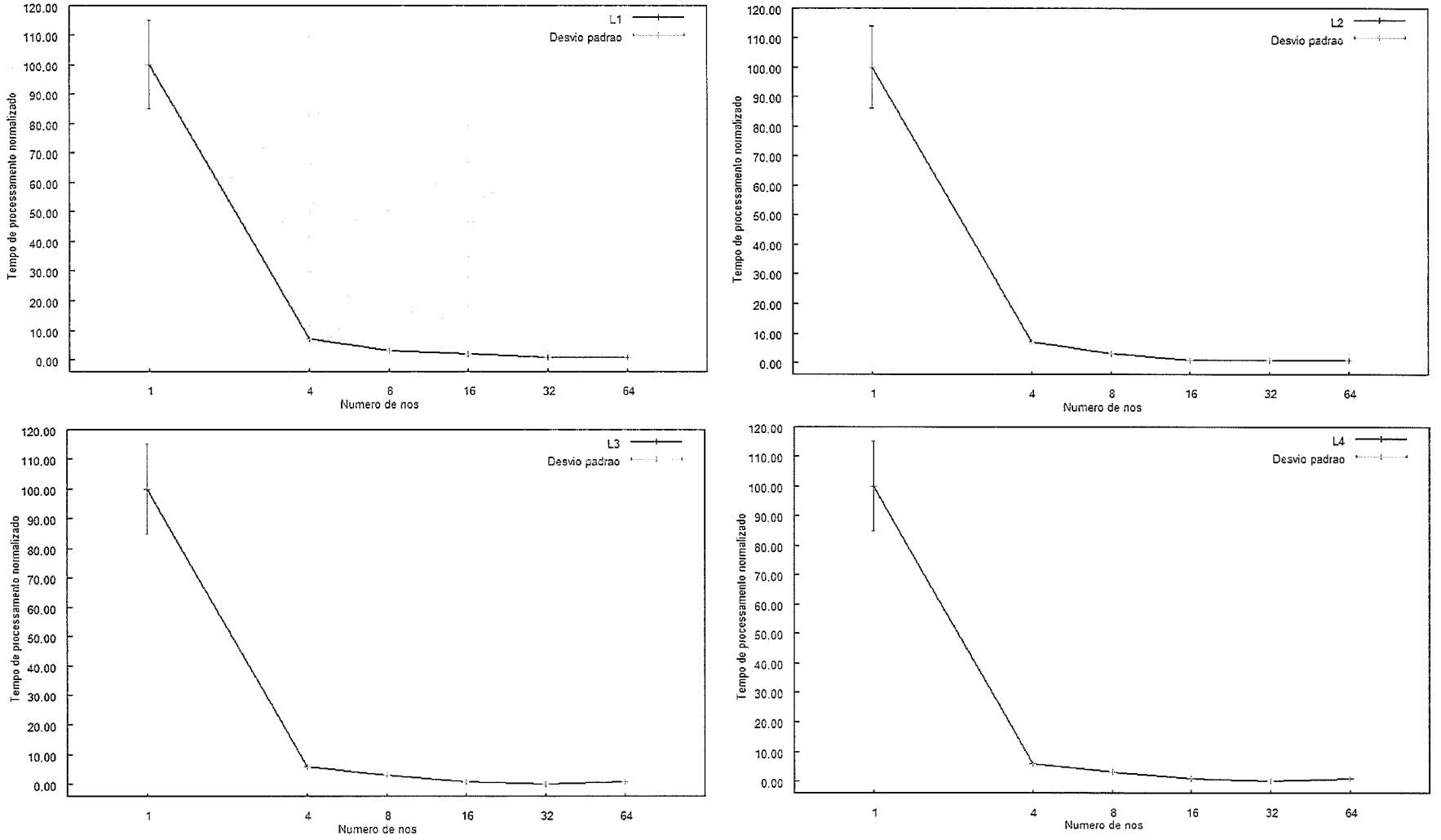
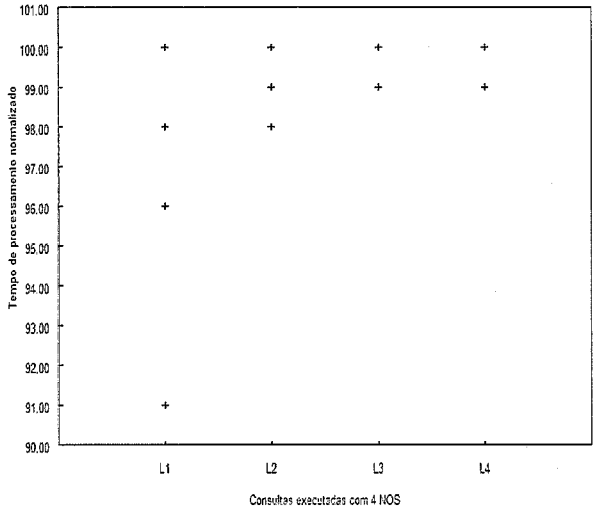
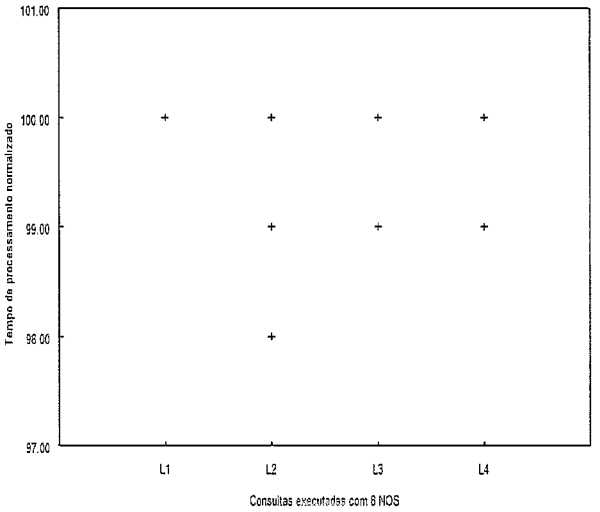
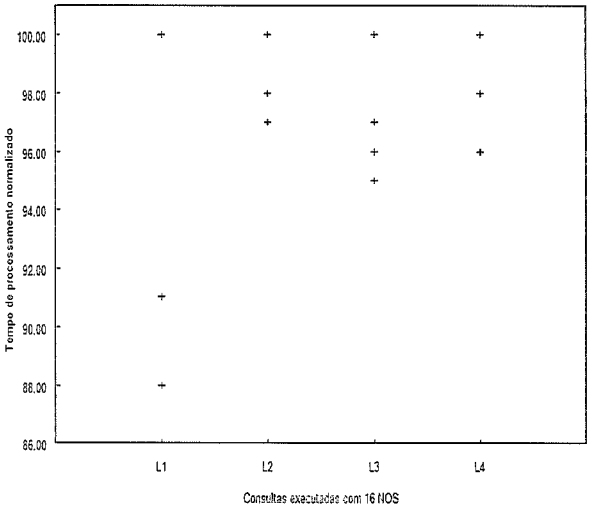
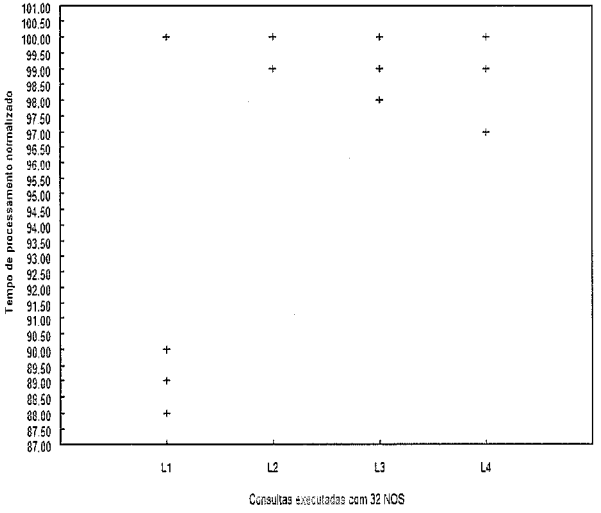


Figura 19: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de força** - BASE PARCIALMENTE REPLICADA – 1 RÉPLICA









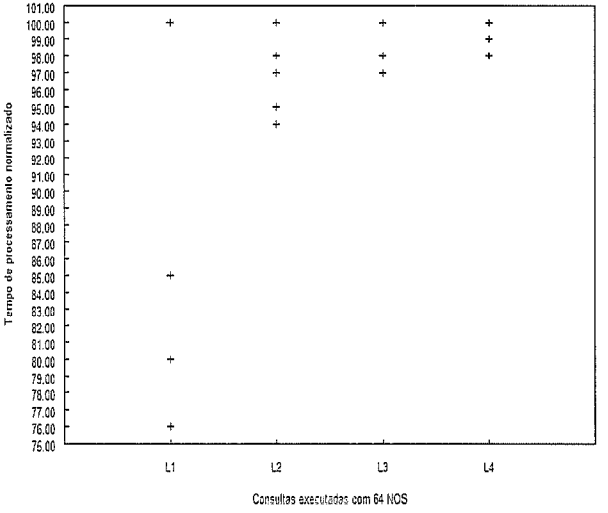


Figura 20: Tempos de execução de cada lote por número de nós - **Teste de força** - BASE PARCIALMENTE REPLICADA – 2 RÉPLICAS

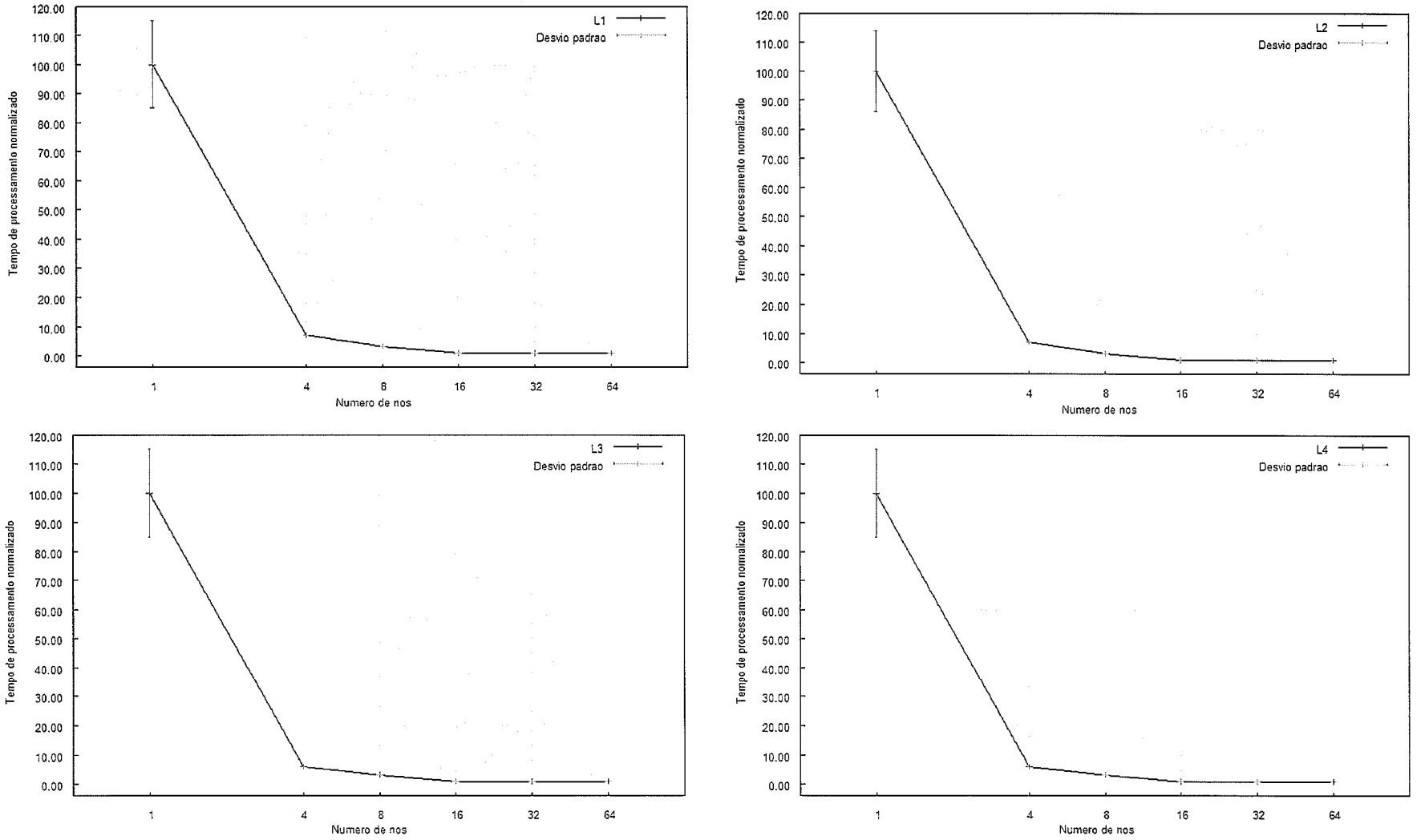


Figura 21: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de força** - BASE PARCIALMENTE REPLICADA – 2 RÉPLICAS

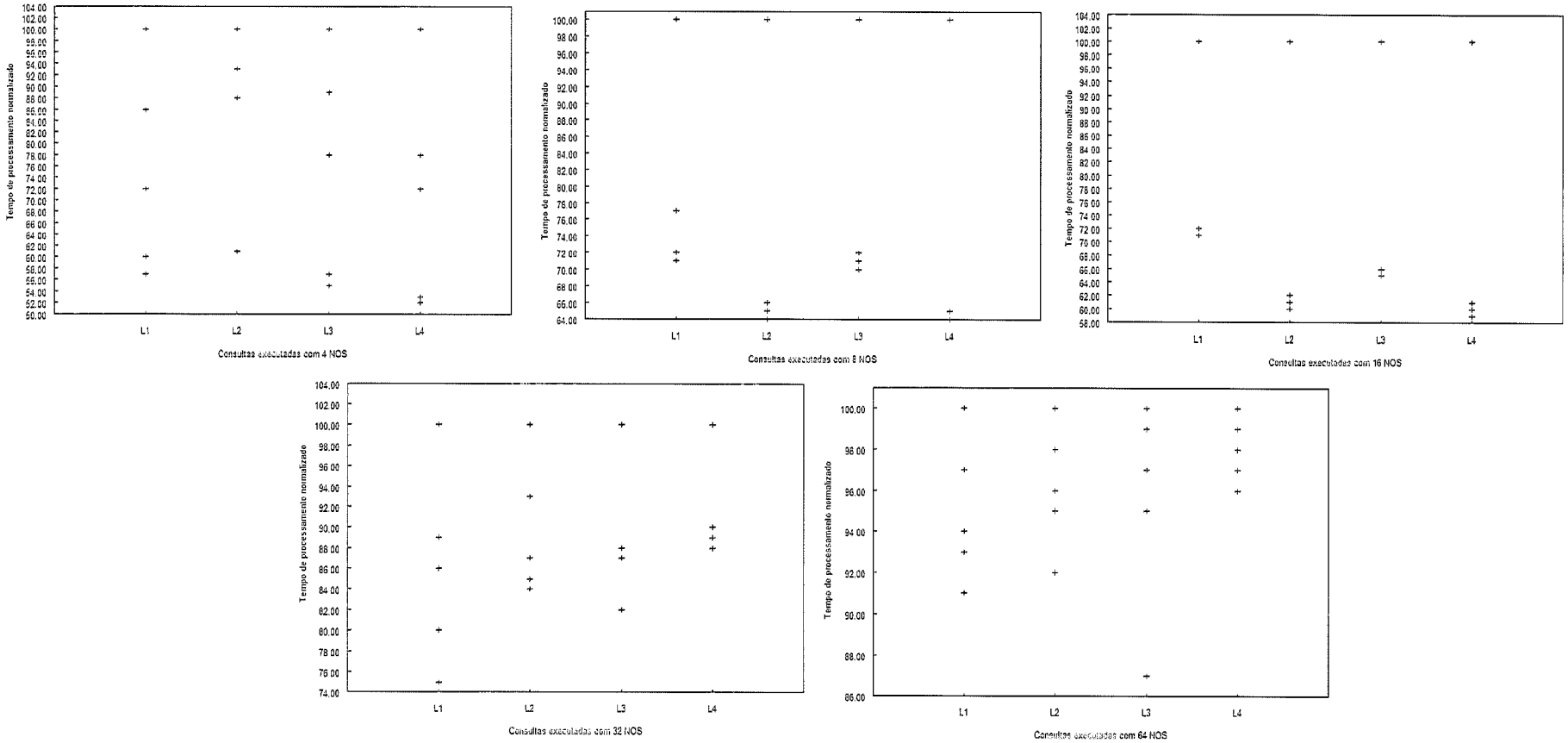


Figura 22: Tempos de execução de cada lote por número de nós - **Teste de carga** - BASE PARCIALMENTE REPLICADA – 1 RÉPLICA

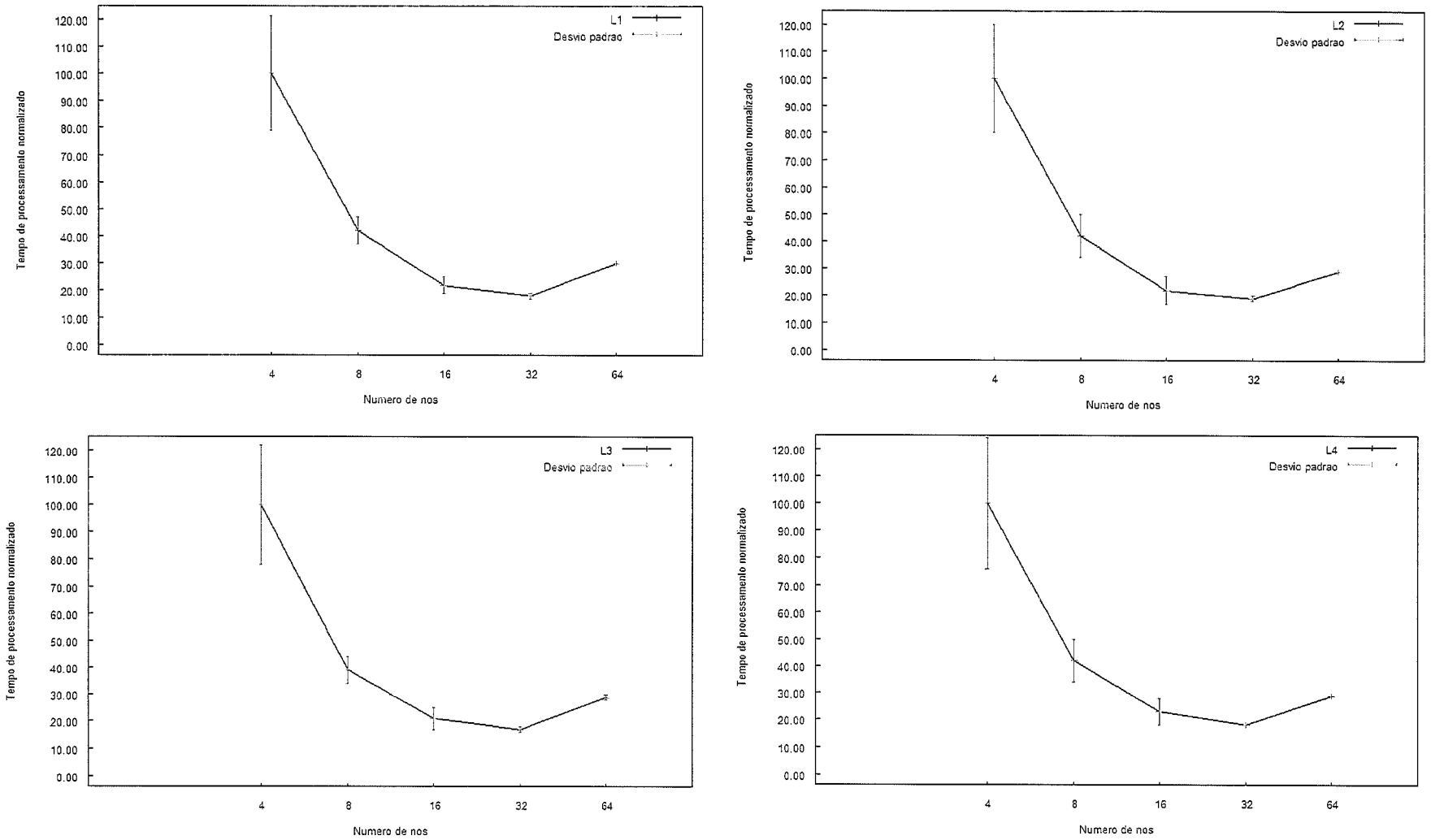


Figura 23: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de carga** - BASE PARCIALMENTE REPLICADA – 1 RÉPLICA

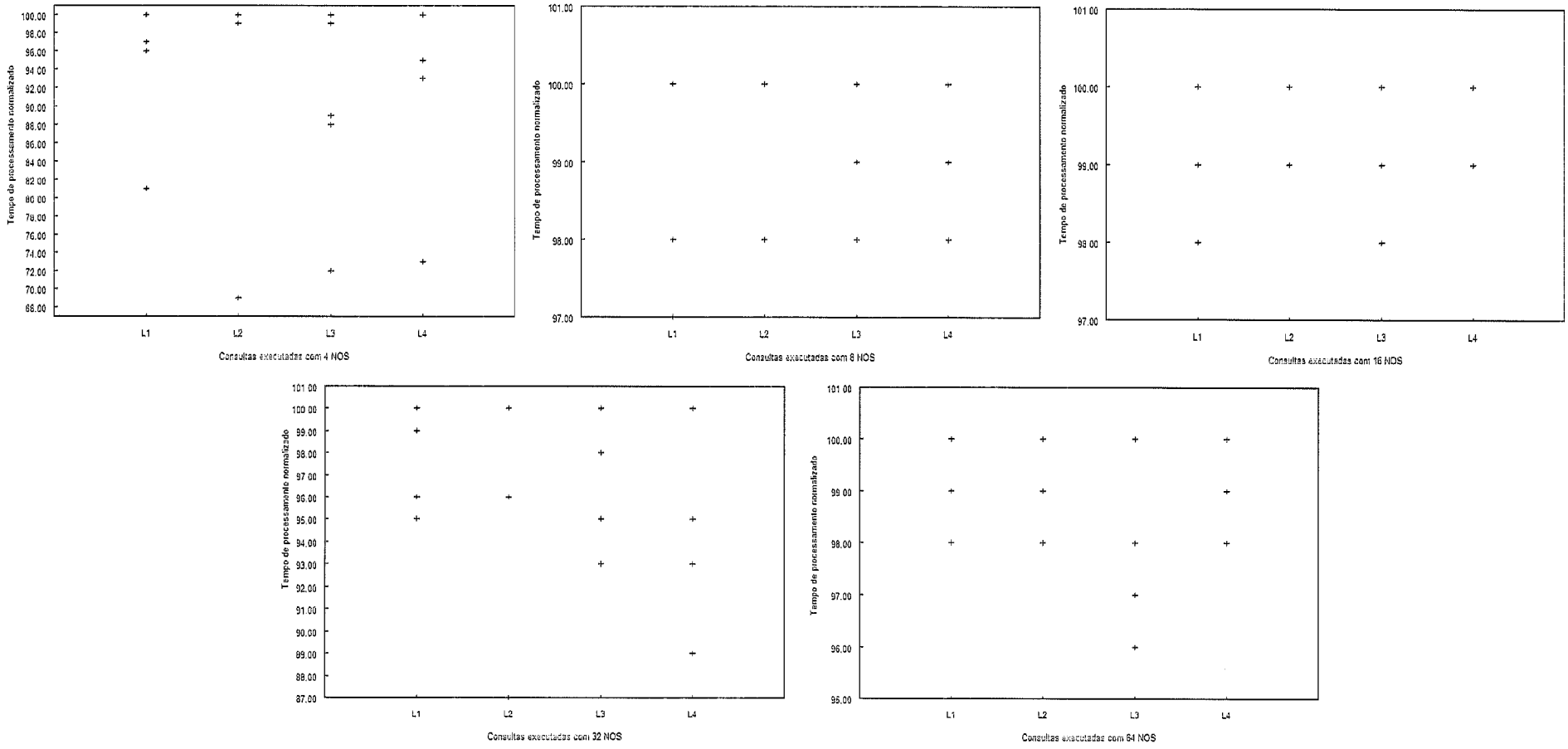


Figura 24: Tempos de execução de cada lote por número de nós - **Teste de carga** - BASE PARCIALMENTE REPLICADA – 2 RÉPLICAS

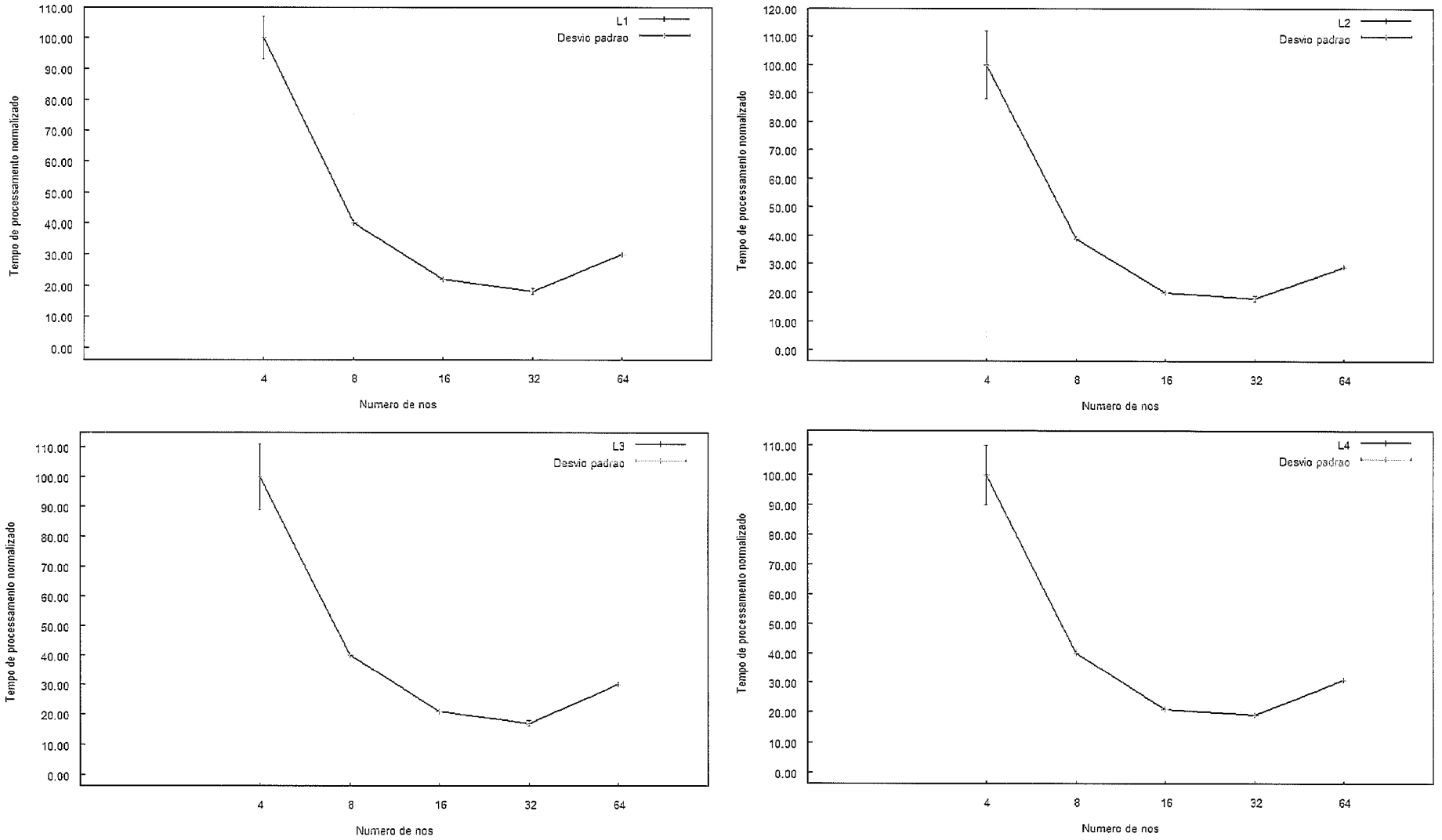


Figura 25: Desvio padrão das médias do tempo de execução – Lotes L1 a L4 – **Teste de carga** - BASE PARCIALMENTE REPLICADA – **2 RÉPLICAS**