

## **Análise no Uso do Controle de Concorrência no Acesso de Dados em MySQL e MariaDB**

**Hiago de Assis Silva<sup>1</sup>, Flávio Ferreira Borges<sup>2</sup>, Esdras Bispo Júnior<sup>2</sup>, Paulo Afonso Parreira Júnior<sup>2</sup>**

<sup>1</sup>Universidade Federal de Goiás – Campus Jataí  
BR 364, Km 193, nº 3800, CEP: 75801-615 – Jataí – GO – Brasil

<sup>2</sup>Departamento de Ciências da Computação  
Universidade Federal de Goiás – Campus Jataí – Jataí, GO - Brasil

hiagoassis@gmail.com, flavio@ufg.br, bispoj@ufg.br,  
pauloafpj@ufg.br

**Abstract.** *This paper presents a research on the performance in use of concurrency control on data access in MySQL and MariaDB, to analyze the behavior of these systems in terms of response time to control and manage this resource. And also show which is the performance gain the MariaDB in relation at MySQL. Thus it's possible to analyse the difference of the performance and verify which they servers testing obtains better gains with different loads.*

**Resumo.** *Este artigo apresenta um trabalho de investigação do desempenho no uso de controle de concorrência no acesso de dados em MySQL e MariaDB, visando analisar o comportamento destes sistemas em termos de tempo de resposta ao controlar e gerenciar esse recurso. E também mostra qual é o ganho de desempenho do MariaDB em relação ao MySQL. Dessa forma, é possível analisar a diferença de desempenho e verificar qual dos servidores testados obtém melhores resultados com diferentes cargas.*

### **1. Introdução**

Com os avanços das organizações e o crescente aumento no volume de dados manipulados tornou-se necessário que os sistemas evoluíssem da mesma forma. Uma série de sistemas específicos foram desenvolvidos para suprir as diversas necessidades de um banco de dados, e essa tecnologia é nomeada Sistema Gerenciador de Banco de Dados (SGBD) [Elmasri *et al.* 2005].

Um SGBD é utilizado para manter controle sobre os dados e também sobre os programas que os acessam. Estes sistemas surgiram devido à necessidade de métodos de gerenciamento computadorizados mais eficazes. Dessa forma, permitem acesso aos dados armazenados em um banco de dados de maneira eficiente e precisa, atendendo as necessidades dos seus usuários [Silberchatz *et al.* 2006].

A comunicação entre usuários e SGBDs é possível graças a um padrão desenvolvido na década de 70, denominado *Structured Query Language* (SQL). A SQL é um padrão para SGBD relacionais e possibilita dentre outras funções, a requisição e manipulação aos dados por meio de consultas e atualizações, bem como a criação de estruturas para o armazenamento de dados (tabelas) [Elmasri *et al.* 2005], [Ramakrishnan *et al.* 2008].

Em geral, SGBDs permitem que várias requisições tenham acesso a um banco de dados ao mesmo tempo, ocasionando concorrência entre as transações<sup>1</sup>. Dessa forma, algum controle é necessário para garantir que transações concorrentes não possam prejudicar umas as outras. Portanto, existem diversos mecanismos de controle de concorrência que devem tratar algumas situações em que uma transação pode resultar em resposta errada, caso tenha sofrido interferência de outras requisições [Date 2003].

A técnica de bloqueio é o principal método utilizado para solucionar a concorrência entre transações [Silberchatz *et al.* 2006]. Normalmente, cada item de dado possui uma variável de bloqueio correspondente. E essas variáveis permitem obter informações sobre operações possíveis de ser aplicadas a estes itens de dados. Existem algumas formas diferentes de realizar bloqueio quanto ao controle de concorrência, entretanto, a aplicação dessas funcionalidades devem ser analisadas com cautela, pois podem implicar em problemas de desempenho, como exemplo, atraso excessivo para atender uma requisição [Elmasri *et al.* 2005].

Contudo, o objetivo deste trabalho é verificar o desempenho dos SGBDs MySQL e MariaDB, quando em uso de controle de concorrência no acesso aos dados, para configuração padrão e também por meio da implementação de bloqueio e desbloqueio (comando *GET\_LOCK()*). Deseja-se verificar como se comportam, em termos de tempo de resposta o controle e gerência deste recurso nestes SGBDs.

A análise de desempenho permite obter métricas da eficiência de produtos, sobre alguma funcionalidade específica [Ciferri 1995]. Para avaliar o desempenho da concorrência nesse trabalho, é usada a técnica de *benchmark*. Segundo Vieira *et al.* (2005), o propósito do uso de *benchmarks* é apresentar formas padronizadas de avaliação do desempenho e confiabilidade de sistemas computadorizados.

Geralmente em *benchmark*, é definido primeiramente um conjunto de testes, os quais podem ser aplicados a diferentes sistemas, desde que apresentem funcionalidades semelhantes. Os resultados colhidos podem então ser analisados por uma metodologia de avaliação definida antecipadamente [Ciferri 1995]. O JMeter possui estas técnicas mencionadas e foi a ferramenta utilizada neste trabalho.

O Apache JMeter é uma ferramenta de código aberto e possui distribuição livre para aplicações de testes de sistemas, possibilitando sua aplicação também em rede distribuída. É baseada em *Java* e passível de extensão devido a *Application Programming Interface* (API) fornecida pela Apache. Após seu desenvolvimento, foi expandido para realizar testes em servidores *File Transfer Protocol* (FTP), servidores de banco de dados, e *JavaServlets* de objetos [Halili 2008].

Todavia, autores como Santos (2008) mostram a importância da realização de testes de desempenho e estresse para avaliar a qualidade de sistemas computacionais com a utilização da ferramenta de *benchmark* JMeter. Em Santos (2008) a avaliação é aplicada a SGBDs com intuito de verificar se existem possíveis perdas de desempenho e até mesmo a não disponibilidade de serviços.

A investigação sobre o desempenho entre MySQL e MariaDB se faz relevante pela preocupação em estar migrando aplicações que utilizavam o MySQL como seu

---

<sup>1</sup> Uma transação é uma unidade lógica de trabalho em banco de dados e envolve diversas operações.

gerenciador de dados para um novo SGBD, especificamente, o MariaDB. Este trabalho pode contribuir em pesquisas que buscam avaliar as características de um banco de dados e definir alterações para adequá-lo a um melhor padrão de funcionamento.

## 2. Trabalhos Relacionados

O objetivo que se tem com esses trabalhos é apresentar as discussões e problemas relacionados à SGBDs, e alcançar maneiras e métodos para realizar uma avaliação de desempenho entre *softwares*. De modo que seja fundamentada em um alto nível de imparcialidade para mostrar a relevância da realização desta análise.

Em Perseguinte (2012), é apresentado o estudo do controle de concorrência com ênfase nas suas características em diferentes arquiteturas de banco de dados. Esse trabalho destaca a importância de manter o controle de concorrência em banco de dados e também como a manipulação de suas técnicas devem ser consideradas caso a caso. Mostra ainda, um estudo sobre o comportamento e evolução dos mecanismos de controle de concorrência, que buscam se adaptar aos pré-requisitos de novas arquiteturas.

O trabalho de Kyoung-Don *et al.* (2007) é um exemplo de como são manipulados sistemas computacionais que enfrentam sobrecarga de trabalho. Nesse trabalho, são abordados servidores de banco de dados sobrecarregados, em especial servidores de tempo real, ditos *real-time database* (RTDB). Para essas aplicações, o tempo de resposta ao (s) requisitante (s) é o ponto chave para seu sucesso.

O autor utiliza um tipo de tecnologia chamada *Chronos*, que pretende aliviar a sobrecarga sobre estes servidores. Esta ferramenta faz uso de algumas técnicas específicas, como exemplo, utiliza-se do bloqueio em duas fases para tentar manter o controle da concorrência sob as aplicações de tempo real utilizadas.

Em Pires *et al.* (2006) é realizado um comparativo de desempenho entre SGBDs de código aberto utilizando o *benchmark Open Source Database Benchmark* (OSDB), com objetivo de analisar as métricas geradas e sugerir possíveis melhorias em seus desempenhos. Este trabalho é útil para a presente avaliação por mostrar a importância de se realizar testes de desempenho em SGBD e como devem ser feitos por meio de ferramentas específicas para esse fim.

O trabalho de Santos *et al.* (2008) apresenta a importância da realização de testes de desempenho e estresse na qualidade dos produtos e também sobre o uso da ferramenta com técnicas de *benchmark* JMeter, de acordo com autor, essa ferramenta é apropriada para esse tipo de teste.

Em Silva (2013) é apresentado uma investigação sobre o desempenho do uso de visão em consultas ao banco de dados, permitindo identificar se o uso deste recurso ameniza a sobrecarga de requisições. Foram utilizados os SGBDs MySQL e PostgreSQL, e por meio dos resultados apresentados, foi possível notar que os dois gerenciadores obtiveram ganho de desempenho nas consultas quando em uso de visão, sendo este ganho foi superior no PostgreSQL. A ferramenta de *benchmark* utilizada por Silva (2013) foi o JMeter.

Por fim, a análise realizada por Schwenke (2012), apresenta os SGBDs MariaDB-5.5.24, MySQL-5.5.25 e Percona Server 5.5.24-26.0. Buscou-se neste trabalho, a realização de testes de desempenho por meio do *benchmark* Sysbench

OLTP. A partir de testes que envolvem operações de leitura e escrita no banco de dados, foram apresentados como resultados os valores de transações por segundo realizadas por estes SGBDs e também o tempo de resposta em que esses gerenciadores atenderam as requisições de diferentes quantidades de usuários enviadas pelo Sysbench.

Indicou-se nesta avaliação, com base nos resultados obtidos, que em relação a operações de leitura, não há muita diferença em termos de desempenho entre estes três SGBDs. Porém, em operações de escrita no banco de dados, afirmou-se que o MySQL e o MariaDB atenderam uma maior taxa de transferência de requisições por segundo.

Enfim, conforme apresentado, há uma necessidade de analisar e comparar tecnologias diferentes sob alguns aspectos específicos. Devem ser realizadas de forma automatizada e metodológica, de forma que certifique relevância e coerência na escolha dos *softwares* adequados para cada caso. Portanto, é evidente a importância da investigação proposta, pois a análise de desempenho retorna diretrizes que podem trazer ganhos a diversos sistemas.

### 3. Metodologia

A metodologia utilizada nessa análise é a *Goal Question Metric* (GQM). Ela indica que para realizar a coleta e avaliação de métricas é necessário primeiramente definir metas. Esse tipo de avaliação permite detalhar um sistema de mensuração, aplicado para um conjunto definido de características e um conjunto de regras que possibilitem interpretar os dados mensurados [Dal'osto 2004].

São detalhados abaixo, os pontos GQM dessa metodologia:

- **Objetivo (*Goal*):** O objetivo deste trabalho é verificar o desempenho do controle de concorrência no MySQL e MariaDB para os comandos SQL *select* e *update* variando o tamanho das consultas e atualizações e dessa forma influenciando diretamente no esforço do SGBD.
- **Questões (*Question*):** Qual é o desempenho em termos de tempo de resposta em que cada SGBD atende requisições de diferentes usuários simultaneamente? Para qual tipo de transação o SGBD tem um melhor desempenho?
- **Métricas (*Metric*):** Linha de 90% dos tempos de resposta fornecidas pelas JMeter e Speedup.

**Linha de 90%:** A métrica linha de 90% fornecida pelo JMeter, é uma medida adequada para testes de desempenho. Ela representa que 90% dos testes apresentaram valores até o tempo de resposta fornecido por essa métrica. Em avaliações, os valores obtidos sobre tempos de resposta sofrem grandes variações, ou seja, resultam em valores muito baixos e valores muito altos na mesma série de dados. Segundo Moffatt (2013) a média aritmética nestes casos pode não apresentar resultados eficientes.

**Speedup:** Ao considerar o impacto de alguma melhoria de desempenho em um sistema, ou de um sistema para outro, normalmente o efeito dessa melhoria é expresso em termos de aumento de velocidade. O *Speedup* (S) é uma expressão que mostra esse ganho de desempenho, considerando como a razão entre o tempo de execução sem a melhoria ( $T_{w0}$ ) para o tempo de execução com a melhoria ( $T_w$ ) [Murdocca 1999]. Para realizar este cálculo, considerou-se o MariaDB como sendo um sistema com melhorias, pelo fato de ser um SGBD recente e que oferece como alternativa ao MySQL, que

consequentemente foi o sistema a ser comparado. Dessa forma, o *Speedup* demonstra em determinadas situações qual é o ganho de desempenho do MariaDB em comparação ao MySQL.

$$S = \frac{T_{W0}}{T_W}$$

#### 4.1. Descrição dos Testes

Os testes foram realizados em um servidor *DELL® PowerEdge 2900*, que utiliza o sistema operacional *CentOS 6.4 final*, com as seguintes configurações: 8 processadores de 2 núcleos *Intel® Xeon E5410 2.33 GHz* (cada núcleo), memória *RAM* de 8 GB, 4 HDs *SATA 250GB 7.2k*, localizado no laboratório de pesquisas em Ciências da Computação da Universidade Federal de Goiás, unidade de Jataí/GO.

A etapa inicial compreendeu na instalação dos bancos de dados MySQL 5.5.33 e MariaDB 5.5.34 no servidor. Para isso, foram utilizadas duas máquinas virtuais (*Oracle VM VirtualBox*), cada uma portando 3 GB de memória *RAM*, e utilizando 3 processadores (6 núcleos). O sistema operacional utilizado nas máquinas virtuais foi o *Debian 7.2. Wheezy*.

A escolha dos SGBD se deve a possibilidade de substituição do MySQL pelo MariaDB, em virtude da aquisição do MySQL por uma empresa que trabalha com softwares proprietários, além do fato de que grandes organizações estão migrando ou analisando migrar para esse SGBD de código livre. Já a escolha do sistema operacional *Debian*, foi devido ao maior reconhecimento da comunidade científica quanto à utilização de sistemas com derivação GNU/LINUX.

Para simular o mais próximo de um ambiente em que SGBDs são utilizados na prática, aplicou-se a ferramenta *apache-jmeter-2.10* em uma máquina cliente conectada via rede ao servidor. Esse computador possui com as seguintes configurações: *Notebook Sony® Vaio*, processador *Intel® i5*, 4 GB de memória *RAM*, HD de 500 GB e utilizando sistema operacional *Debian 7.2. Wheezy*.

Para a realização dos testes, foi adotada uma base de dados de gerenciamento escolar, que contém 103 tabelas e 97 relacionamentos e com tamanho de aproximadamente 135MB, já igualmente predefinida para os dois SGBDs. Foram construídos seis comandos SQL para os dois servidores, sendo eles três *select* e três *update*, alguns destes possuem várias interações entre as entidades, de modo que exija elevado processamento de todo o sistema.

#### 4.2. Carga de Trabalho

Buscou-se com as cargas de trabalho analisar o desempenho quanto à concorrência quando em uso de requisições de cargas que exigem pouco do sistema e também com cargas que necessitam de grande recurso computacional. Elas foram aplicadas aos dois SGBDs na seguinte ordem:

1. Foram executadas *select* de carga e *update* de carga baixa com 4, 8, 16, 32, 64, 128, 256, 512 e 1024 usuários simultâneos, no MariaDB e posteriormente no MySQL em configuração padrão.

2. Alterado o valor da variável *innodb\_lock\_wait\_timeout*, foi realizado um *update* de carga alta no MariaDB e posteriormente no MySQL com 4, 8, 16, 32, 64, 128, 256, 512 e 1024 usuários simultâneos.

3. Foram executados *select* e *update* de carga alta com o uso de *GET\_LOCK()* para o MariaDB e posteriormente no MySQL com 4, 8, 16, 32, 64, 128, 256, 512 e 1024 usuários simultâneos.

4. Por último, foram executados *select* e *update* de carga baixa, com o *pool-of-threads* ativo no MariaDB também com 4, 8, 16, 32, 64, 128, 256, 512 e 1024 usuários simultâneos.

#### 4. Resultados Obtidos

Foram realizados diferentes testes, obtendo resultados da linha de 90% fornecida pelo JMeter. Com esses tempos de resposta, foram calculados a porcentagem da diferença entre os resultados para MySQL e MariaDB para cada tipo de teste. Também foram calculados os *Speedup* para verificar e se houve ganho de desempenho do MariaDB em relação ao MySQL para os mesmos casos.

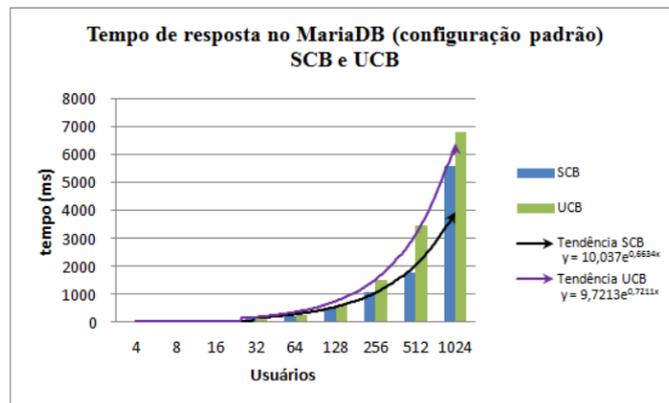


Figura 1. Tempo de resposta no MariaDB com a configuração padrão utilizando *select* com carga baixa e *update* com carga baixa.

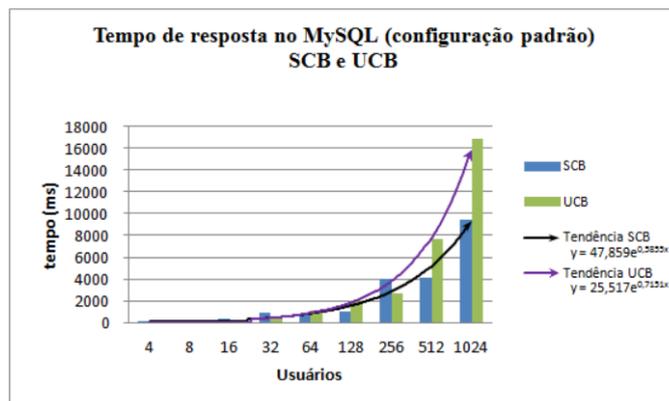


Figura 2. Tempo de resposta no MySQL com a configuração padrão utilizando *select* com carga baixa e *update* com carga baixa.

A Figura 1 e Figura 2 mostram exemplos de alguns dos testes realizados, apresentando os tempos de resposta com diferentes quantidades de usuários para MySQL e MariaDB em configuração padrão. Neste caso, o MariaDB obteve resultados mais satisfatórios, respondendo mais rapidamente as requisições que exigem pouco

esforço computacional. Enquanto que para requisições com carga maior, que exigem mais de todo o sistema, o MySQL obteve uma pequena vantagem.

Por meio dos testes realizados, observou-se que o MariaDB tem desempenho superior em termos de tempos de resposta quando exposto a situação de concorrência para simples consultas e atualizações em tabelas. Por outro lado, para uma alta carga de trabalho o MySQL se mostrou pouco superior na maioria dos testes. Em relação à utilização do recurso *GET\_LOCK()*, a diferença entre os SGBDs foi mínima, embora o tempo de resposta em ambas tenha aumentado significativamente. Por fim, fez-se o uso do recurso *pool\_of\_threads* nas configurações do MariaDB, mostrando-se eficiente para quantidades iguais ou superiores a 256 usuários, melhorando o desempenho. É importante ressaltar que este recurso não foi possível ser utilizado nos testes com o MySQL, pois o mesmo é ofertado somente nas versões com pagamento de licenças comerciais.

## 5. Conclusão e Trabalhos Futuros

Tendo em vista os aspectos observados, conclui-se que o MariaDB pode ter um melhor desempenho em sistemas que realizem simples transações no banco de dados, mesmo que com um número elevado de usuários acessando a base de dados simultaneamente. Este desempenho pode ser aprimorado com o uso do *pool of threads* para uma quantidade maior de clientes concorrentes. Para sistemas que realizem requisições de carga de trabalho elevada, conclui-se que tanto o MariaDB e MySQL, por possuírem desempenhos bastante próximos, provavelmente terão comportamentos semelhantes.

As informações apresentadas são úteis para escolha e migração entre estes SGBDs, pois percebe-se que existem vantagens e desvantagens da utilização de cada quanto a concorrência no acesso de dados.

Como trabalhos futuros pretende-se analisar o desempenho destes SGBDs para outros tipos de transações e também verificar quais os fatores que resultam em diferenças de desempenhos. Outro trabalho pretendido é analisar a concorrência entre vários usuários realizando diversos tipos de transações ao mesmo tempo. Analisando também quais as características de SGBD influenciaram no tempo de respostas.

Também pretende-se realizar um estudo sobre outras técnicas ou recursos que contribuem na otimização dos tempo de resposta em situações de concorrência. Dessa forma, determinar quais possuem um melhor desempenho e para qual caso cada técnica será apropriada.

## References

- CIFERRI, Ricardo R. Um Benchmark Voltado à Análise de Desempenho de Sistemas de Informações Geográficas. 1995. 192 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Campinas, Campinas, SP, 1995.
- DAL’OSTO, Fábio. Método para Avaliação de Ambientes de Desenvolvimento de Software Combinando CMM e GQM. 2003. 110 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, RS. 2003.
- DATE, C. J. Introdução a Sistemas de Banco de Dados. 4. ed. Rio de Janeiro: Editora Elsevier, 2003.

- ELMASRI, Ramez; NAVATHE, Shamakant B. *Sistemas de Banco de Dados*. 4. ed. São Paulo: Editora Pearson Addison Wesley, 2005.
- HALILI, Emily H. *Apache JMeter: A Practical Beginner's Guide to Automated Testing and Performance Measurement for your Websites*. 1. ed. Birmingham: Packt publishing, 2008.
- MOFFATT, Robin. Performance and OBIEE – part VI – Analysing results. Disponível em <<http://www.rittmanmead.com/2013/03/performance-and-obiee-analysing-results/>> Acesso em: 01 jan. 2014.
- MURDOCCA, Miles J.; HEURING, Vicent P. *Principles of Computer Architecture*. 1. ed. Prentice Hall, 1999.
- KYOUNG-DON, K.; SIN, P. H.; JISU, O., SON, S. H. A Real-Time Database Tested and Performance Evaluation. 3th IEEE International Conference. Embedded and Real-Time Computing Systems and Applications, 2007.
- PERSEGUINE, Vanessa Ravazzi. *Análise do controle de concorrência em diferentes tecnologias de banco de dados*. 2012. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação. 2012.
- PIRES, Carlos E. S.; NASCIMENTO, Rilson O.; SALGADO, Ana C. *Comparativo de Desempenho entre Banco de dados de Código Aberto*. Escola Regional de Banco de Dados. Anais da ERBD06. Porto Alegre: SBC, 2006.
- RAMAKRISHNAN, Raghu; GEHRKE, Johannes. *Sistemas de Gerenciamento de Banco de Dados*. 3. ed. Rio de Janeiro: Editora McGraw-Hill, 2008.
- SANTOS, Ismayle S.; NETO, Pedro A. S. *Automação de Testes de Desempenho e Estresse com o JMeter*. Livro-texto dos minicursos do II ERCEMAPI – Escola Regional de Computação, São Luiz, MA. Cap 7, p 151-176, 2008.
- SCHWENKE, Axel. 5.5 Series Sysbench OLTP Results. Disponível em: <<https://blog.mariadb.org/5-5-series-sysbench-oltp-results/>> Acesso em: 24 jan. 2014.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistemas de Banco de Dados*. 5. ed. São Paulo: Editora Elsevier, 2006.
- SILVA, Ricardo C. *Benchmark em Banco de Dados Multimídia: Análise de Desempenho em Recuperação de Objetos Multimídia*. 2006. 64 f. Dissertação (Mestrado em Informática) – Universidade Federal do Paraná, Curitiba, PR, 2006.
- SILVA, W. J.; SILVA, H. de A.; BORGES, F. F.; *Desempenho do Uso de Visões nas Consultas em Diferentes Servidores de Banco de Dados Relacionais*. In: EnAComp, X Encontro Anual de Computação, Universidade Federal de Goiás (Câmpus Catalão), Catalão – Goiás, Anais, 2013, 89-96.
- VIEIRA, Marco.; DURÃES, João.; MADEIRA, Henrique. *Especificação e Validação de Benchmarks de Confiabilidade para Sistemas Transaccionais*. IEEE Latin America Transactions, Jun. 2005.