Relatório de Trabalhos
Projecto: "StrongRep"
Período: 01/12/2002 a 30/11/2003

Alfrânio Tavares Correia Júnior

19 de Janeiro de 2004

Este relatório descreve as actividades de investigação científica, realizadas durante o período de 01 de Dezembro de 2002 a 30 de Novembro de 2003, no âmbito do projecto "StrongRep - Strongly Consistent Replicated Databases in Geographically Large-Scale Systems", referência FCT POSI/CHS/41285/2001, sob a supervisão do Professor Doutor Rui Carlos Oliveira.

1 Situação Inicial

O objectivo do projecto StrongRep é analisar e propor alternativas para as questões relacionadas a replicação de base de dados em larga escala [1], sem abrir mão de um critério de coerência forte. Basicamente, os protocolos de replicação tradicionais não apresentam um bom desempenho em ambientes de larga escala, originando um grande número de mensagens entre os *sites* e um grande número de *deadlocks*. Para contornar isso, pretende-se explorar a crescente promessa de escalabilidade e desempenho sugerida pela técnicas de replicação de base de dados com protocolos de comunicação em grupo e os esforços para implementar e optimizar protocolos de difusão que tiram partido da topologia de rede e da semântica das aplicações.

Diversas pesquisas na área de replicação de banco de dados com protocolos de comunicação em grupo tem sido desenvolvidas [2, 3, 4, 5, 6, 7], e entre elas encontra-se uma proposta conhecida como *Database State Machine* ou simplesmente *DBSM* [2]. Esta proposta proporciona uma execução local optimista das transacções, permitindo que interacções com transacções concorrentes ocorram apenas após a solicitação do término da transacção pelo cliente. Essas interacções são geridas por um processo determinístico que ocorre em dois passos. No primeiro passo, a transacção é propagada utilizando um protocolo de difusão atómica, o que garante que todas as réplicas recebem a transacção na mesma ordem. No segundo passo um procedimento denominado de certificação garante que as alterações realizadas pelas transacções sejam aplicadas por todas as cópias, se as transacções correntes forem serializáveis, ou as transacções ordenas posteriormente que violam o critério de serialização são encerradas [8].

Neste cenário, o objectivo inicial proposto para o bolseiro envolvia de uma maneira geral: (*i*) analisar em detalhe os conceitos da *DBSM*, (*ii*) estudar o uso de replicação de base de dados com protocolos de comunicação em grupo no âmbito do projecto *GLOBDATA*, (*iii*) analisar e implementar um benchmark padrão de mercado para avaliar a *DBSM*, (*iv*) analisar a *DBSM* num cenário de replicação parcial, extendendo o trabalho desenvolvido no projecto *Escada*.

A descrição completa das tarefas propostas inicialmente pode ser vista a seguir:

- **Tarefa 01**
  Estudo detalhado da teoria de serialização [8], utilizada para garantir a consistência na execução de transacções concorrentes. Esse conceito representa a base de todo o controlo de concorrência em base de dados, sendo de suma importância para o bom entendimento e realização das outras tarefas.

- **Tarefa 02**
  Estudo detalhado dos conceitos da *DBSM* [2] e a materialização das ideias apresentadas para um banco de dados de mercado. Especificamente, a materialização consiste em verificar as prováveis dificuldades e problemas que podem ser encontrados para implementar os conceitos apresentados na *DBSM*.

- **Tarefa 03**
  Estudar como o *G*LOB*D*ATA [9] utiliza os conceitos de replicação de base de dados com protocolos de comunicação em grupo.

- **Tarefa 04**
  Estudar e implementar um benchmark [10, 11] padrão de mercado para avaliar o desempenho da *DBSM*.

- **Tarefa 05**
  Extender o sistema *Escada* [1], permitindo o uso transparente a uma base parcialmente replicada, onde normalmente se faz necessário o acesso a diversos *sites* para atender as solicitações dos clientes. Além disso, avaliar o impacto da replicação parcial quando o acesso a diversos sites torna-se frequente, e sugerir mecanismos que possam reduzir ou minimizar esse impacto.

2 Desenvolvimento

2.1 Conceitos de Serialização

O estudo inicial da teoria da serialização ocorreu normalmente com havia sido planeado e importantes conceitos foram extraídos e utilizados durante todo o processo de pesquisa. Esses conceitos definem o critério de coerência desejado.

2.2 DBSM

Na *DBSM*, o acrónimo de *Database State Machine*, encontra-se uma proposta de replicação de base de dados utilizando técnicas de comunicação em grupo. Essa proposta representa uma combinação das ideias da replicação activa [12], onde os participantes desenvolvem as actividades de maneira determinística, com o uso de actualizações deferidas [8].

Em outras palavras, durante a execução, as transacções são processadas localmente de maneira optimista, não havendo nenhum tipo de interacção com transacções concorrentes em *sites* remotos. O cliente, ao solicitar o encerramento da transacção, inicia um processo de terminação que consiste em: (*i*) propagar as alterações para todas as cópias, atribuindo-lhe uma ordem pelo uso de protocolos de difusão atómica, (*ii*) e em seguida analisar possíveis ocorrências de conflitos entre transacções concorrentes, num processo denominado de certificação. Aonde a ordem fornecida pela difusão atómica é utilizada para determinar a transacção que deve ser encerrada ou não em caso de conflito.

Ocorre o conflito entre duas transacções t e t', se e somente se: t e t' são concorrentes e possuem operações que acedem a mesma informação e pelo menos uma delas é uma operação de escrita. Portanto, se uma transacção t não possui conflito com outras transacções concorrentes, suas alterações são aplicadas deterministicamente em todas as réplicas. E caso existam conflitos, a transacção t é encerrada sem efectivar as alterações. O leitor pode facilmente observar que o determinismo é alcançado com o uso da difusão atómica que garante a mesma ordem para a transacção t em todos as réplicas e pelo processo de certificação.

Contudo, percebe-se que a *DBSM* apresentada em [2] é altamente abstracta e algumas questões surgem naturalmente em consequência dessa abstracção. As primeiras questões estão relacionados a propagação do resultado de operações de leitura das transacções: (*i*) "Por que propagar as informações lidas por uma transacção se elas não proporcionam mudanças de estado ?" (*ii*) "Como determinar as informações lidas, se isso pode ter variações com a estrutura e optimizações do banco como o uso de índices e estatísticas ?" (*iii*) "O tamanho e a quantidade

[1] Projecto desenvolvido pelo Grupo de Sistemas Distribuídos da Universidade do Minho, cuja referência é FCT POSI/CHS/33792.

das informações lidas pode prejudicar a propagação de uma transacção, como resolver esse problema ?". Para a primeira pergunta, basta observar a definição de conflito estabelecida pela *DBSM* e verificar que as informações lidas são utilizadas para garantir o critério de coerência forte desejado. No que diz respeito a segunda pergunta, em [13] apresenta-se uma definição para as informações lidas e um algoritmo determinístico para sua extracção. De acordo com [13], as informações lidas representam o menor conjunto possível diferente do vazio que permite a re-execução da transacção produzindo o mesmo resultado. Com relação a terceira pergunta, pode-se definir um valor limite por tabela a partir do qual não se transmitem informações lidas, mas sim uma referência para a mesma [13]. Ou pode-se não enviar as informações de leitura, centralizando a certificação no *site* responsável pela comunicação com o cliente e adicionando um passo ao protocolo de terminação que corresponde ao aviso de sucesso ou falha da certificação centralizada [4, 14].

A último questão refere-se a garantia da serialização de acordo com [8]. É de conhecimento vulgar que para se obter execuções equivalentes a execuções seriais precisa-se introduzir também o uso de índices [8, 15] e em nenhum momento é apresentado isso em [2]. Além do mais, durante as pesquisas realizadas não se encontrou nenhum estudo que apresentasse o tema no contexto da *DBSM*, apenas uma alusão ao uso de índices em [14].

2.3 Conceitos do GLOBDATA

O GLOBDATA [9] é um projecto que tem como objectivo desenvolver e implementar um *middleware* capaz de fornecer a abstracção de um repositório global de objectos. Diversas instituições académicas estão envolvidas e participam deste projecto, entre elas a Faculdade de Ciências da Universidade de Lisboa (FCUL), parceiro no StrongRep. COPLA é o nome da ferramenta que pretende materializar todos as premissas estabelecidas para o GLOBDATA e de uma maneira geral e sucinta é responsável por um acesso transaccional independentemente da localização a objectos persistentes geograficamente distribuídos.

O intuito deste ponto do relatório restringe-se a apresentação e análise do protocolo de replicação utilizado pela ferramenta COPLA. Portanto, pode-se definir que estão disponíveis dois protocolos de replicação, acedidos por uma única interface e sendo a escolha determinada por um compromisso entre resiliência e desempenho. Os protocolos são assim classificados de acordo com [16]: (*i*) protocolo sem fase de votação e (*ii*) protocolo com fase de votação.

No protocolo sem fase de votação, segundo [9] tem-se uma adaptação da *DBSM* com o uso de controle de concorrência com versões [8]. Percebe-se que as mudanças dizem respeito a informação manipulada, pois no COPLA as informações são objectos que possuem um identificador, um número de versão e um estado que identifica se a cópia do objecto está ou não actualizada. O número de versão é inicializado com zero sempre que um novo objecto é criado e é incrementado todas as vezes que ocorre uma alteração. Portanto, duas transacções t e t' possuem conflito se forem concorrentes e t' tiver lido um objecto com uma versão v_o e a versão do objecto na base de dados $v_{o'}$ é maior que v_o.

Os passos do algoritmo utilizado são apresentados logo abaixo:

- Todas as transacções são executadas localmente e de uma maneira optimista.
- Quando o término da transacção é requisitado pelo cliente, o conjunto de objectos lidos e sua versão, e o conjunto de objectos escritos são enviados para todas as cópias utilizando um primitiva de difusão atómica.
- Se houver conflito da transacção de acordo com o que foi definido acima a transacção é finalizada, caso contrário as alterações são aplicadas.

No protocolo com fase de votação, uma adaptação de [4, 14], apenas as informações alteradas são enviadas para a replicação e como o *site* onde a transacção foi iniciada é o único local que possui as informações de leitura, apenas ele pode proceder com a detecção de conflitos, ocasionando um passo adicional ao protocolo anterior que é a confirmação do término da transacção.

Por último, uma optimização denominada de *deferred update* é proposta aos dois protocolos anteriores, optando-se por uma menor tolerância a faltas em troca de um melhor desempenho. Nessa optimização não são enviados os objectos alterados propriamente ditos, mas apenas identificadores, pois as alterações permanecem no servidor

onde a transacção foi iniciada até serem requisitadas. Para que essa optimização funcione, quando um objecto é requisitado para leitura, é verificada se a cópia local está actualizada e caso não esteja a versão mais recente é recuperada.

2.4 Benchmark

Optou-se por estudar e implementar os benchmarks TPC-C [10] e TPC-W [11], actualmente padrões para avaliação *OLTP* (*Online Transaction Processing*). O TPC-C procura simular um ambiente de uma empresa de vendas, com pontos de distribuição espalhados geograficamente. O TPC-W procura simular um ambiente Web de vendas de publicações.

O objectivo com o uso desses padrões era realizar testes com carga mais realista, permitindo uma maior fiabilidade nos resultados obtidos e uma melhor comparação com o desempenho de sistemas comerciais. Contudo, durante os estudos e implementação, percebeu-se que a percentagem de leituras do TPC-W era muito superior a percentagem de escrita, favorecendo o desempenho da *DBSM*. Portanto, para obter resultados mais significativos, optou-se por utilizar apenas a implementação TPC-C.

Deparou-se em seguida com um outro problema. O mecanismo inicial de utilização do benchmark consistia na colecta de dados de um ambiente real, o que dificultava as avaliações, pois os recursos (processador, memória, disco) não eram suficientes para realizar a colecta e em consequência a amostragem obtida não era significativa. Além disso, o factor tempo de colecta dificultava os trabalhos. Portanto, optou-se por uma solução na qual o benchmark proposto alimenta directamente o sistema a ser avaliado.

Os resultados dos testes e uma análise detalhada do uso do TPC-C para avaliação da *DBSM* pode ser encontrada em [13].

2.5 Extensões do Escada

No intuito de permitir a execução distribuída de transacções num ambiente no qual a base de dados encontra-se fragmentada e parcialmente replicada para reduzir os custos de uma replicação total, adicionou-se ao *Escada* um mecanismo para execução de subtransacções.

Infelizmente, como a fragmentação da base de dados está directamente ligada a semântica da aplicação e condicionada ao particionamento realizado pelo administrador caso não haja algum mecanismo automático para isso, pode-se ter um impacto altamente negativo em todo o sistema, em situações em que se faz necessário o constante acesso a *sites* remotos. No intuito de minimizar esse problema explorou-se algumas características da *DBSM*, nomeadamente a execução optimista e a difusão atómica para construir um mecanismo de cache semântico.

3 Ponto de Situação Actual

- **Tarefa 01**
  Realizada completamente.
- **Tarefa 02**
  Realizada parcialmente pois ainda se faz necessário um estudo detalhado dos protocolos de difusão atómica para redes de larga escala.
- **Tarefa 03**
  Realizada completamente.
- **Tarefa 04**
  Realizada completamente.
- **Tarefa 05**
  Está em fase final e terá como resultado a dissertação de mestrado do bolseiro.

4 Conclusões

Com a realização deste trabalho pode-se entender a proposta da *DBSM* e os pontos críticos para a sua implementação:

- Informações Lidas - Os problemas relacionados ao envio das informações lidas, isto é, o seu tamanho e o impacto no consumo de largura de banda e consequentemente no desempenho da *DBSM*. Para contornar esse problema são propostas duas soluções: (*i*) envio de uma quantidade limite de informação a partir do qual se indica que toda a relação foi acedida [13]; (*ii*) não envio das informações e uso de um passo adicional para confirmação do fim da transacção [4, 14].
- Índices - A utilização de índices para garantir as propriedades de serialização.
- Protocolos de Difusão - O uso de protocolos de difusão optimizados para redes de larga escala.
- Replicação Parcial - Exploração da replicação parcial, sem esquecer dos impactos negativos que podem ocorrer quando a fragmentação realizada gera uma quantidade excessiva de subtransacções. E a possibilidade de utilização de mecanismos como o cache semântico para reduzir esse impacto.

Referências

[1] Jim Gray, Pat Helland, Patrick O'Neil, and Dennis Shasha. The dangers of replication and a solution, 1996.

[2] F. Pedone. *The Database State Machine and Group Communication Issues*. PhD thesis, Département d'Informatique, École Polytechnique Fédérale de Lausanne, 1999.

[3] M. Patino-Martinez, R. Jimenez-Peris, B. Kemme, and G. Alonso. Scalable Replication in Database Clusters. In *Proc. of Distributed Computing Conf., DISC'00. Toledo, Spain*, volume LNCS 1914, pages 315–329, October 2000.

[4] Bettina Kemme and Gustavo Alonso. A suite of database replication protocols based on group communication primitives. In *International Conference on Distributed Computing Systems*, pages 156–163, 1998.

[5] Y. Amir, D. Dolev, P. Melliar-Smith, and L. Moser. Robust and efficient replication using group communication, 1994.

[6] G. Alonso. Partial database replication and group communication primitives, 1997.

[7] A. Sousa, F. Pedone, R. Oliveira, and F. Moura. Partial replication in the database state machine. In *IEEE International Symposium on Network Computing and Applications*, pages 298–309, Cambridge, MA, October 2001. IEEE Computer Science.

[8] Vassos Hadzilacos Philip A. Bernstein and Nathan Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[9] L. Rodrigues, H. Miranda, R. Almeida, J. Martins, and P. Vicente. Strong replication in the globdata middleware, 2002.

[10] Transaction Processing Performance Council (TPC). Tpc benchmarkc standard specification revision 5.0 february 26, 2001, 2001.

[11] Transaction Processing Performance Council (TPC). Tpc benchmarktm w (web commerce) specification version 1.7, oct 11,2001, 2001.

[12] R. Guerraoui and A. Schiper. Fault-tolerance by replication in distributed systems. In *Reliable Software Technologies - Ada-Europe'96*, pages 38–57. Springer-Verlag, 1996.

[13] "A. Souza, J. Pereira, L. Soares, A. Correia Jr, L. Rocha, R. Oliveira, and F. Moura". Evaluating performance of the database state machine (dbsm). Technical report, Universidade do Minho, Departamento de Informática, 2003.

[14] B. Kemme and G. Alonso. Don't be lazy, be consistent: Postgres-R, a new way to implement database replication. In *Proceedings of 26th International Conference on Very Large Data Bases (VLDB 2000)*, pages 134–143. Morgan Kaufmann, 2000.

[15] Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, and Patrick O'Neil. A critique of ANSI SQL isolation levels, 1995.

[16] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso. Database replication techniques: a three parameter classification. In *Proceedings of 19th IEEE Symposium on Reliable Distributed Systems (SRDS2000)*, Nürenberg, Germany, 2000. IEEE Computer Society.