Sincronização de bancos de dados distribuídos utilizando Snapshot Isolation

Fabricio Santos da Silva

Universidade Federal do Paraná (UFPR) – Departamento de Informática

Curitiba – Paraná – Brazil

fabricioss@inf.ufpr.br

Abstract. *This paper shows how the behavior of data replication algorithm is during his execution in database distributed system whit snapshot isolation as concurrency control policy. This paper wants to simulate a distributed database system and shows how the algorithm works in artificial conditions of reality.*

Resumo*. Este artigo apresenta como é o comportamento de um algoritmo de replicação de dados em um sistema de banco de dados distribuídos que utilize o conceito de snapshot isolation. O objetivo deste trabalho é simular um ambiente de bancos de dados distribuídos e demonstrar como o algoritmo funciona sobre aspectos de reais de um sistema de banco de dados*

1. Introdução

Bancos de dados há muito tempo são ferramentas úteis na tarefa de armazenamento e busca de informação. Estes sofreram ao longo de sua existência e uso uma variada série de transformações, onde novos modelos para a estrutura de armazenamento foram propostos e adotados (bancos de dados relacionais, banco de dados objeto-relacionais e outros), alterações na forma de armazenamento e busca da informação (criação de índices, possibilidade de particionamento de tabelas, manipulação de parâmetros de uso de memória, etc) proporcionados pela melhoria dos sistemas computacionais como um todo.

Uma das modificações que ganham mais atenção nos últimos tempos em todo o mundo computacional é a possibilidade de distribuição de recursos para armazenamento e processamento entre diversas máquinas. A força que este campo da computação ganhou nos últimos anos se deve ao baixo custo para aquisição de equipamentos e a grande difusão da internet, hoje estes recursos passam a fazer parte do interesse de uso por pequenas e médias empresas e não somente das grandes corporações e meios acadêmicos.

Como toda ferramenta de ponta, os Sistemas de Gerenciamento de Bancos de Dados (SGBDs) também estão sofrendo modificações para distribuir os seus recursos. Bancos de dados distribuídos tem sido abordados com o intuito de tratar os dados que estão distribuídos geograficamente e ou por razões administrativas [4][3]. Assim ganha-se em

diversos aspectos como disponibilidade da informação, maior velocidade de acesso aos dados, mas perde-se nos aspectos que dizem respeito a gerencia destes dados, ou seja, manter um estado global entre todas as SGBDs que compõem a rede de forma consistente.

Recentemente, com o avanço das pesquisas na área de sistemas distribuídos, novos modelos para manter o estado global de um sistema de bases de dados distribuídos estão sendo propostos. Este trabalho apresenta um das alternativas propostas e a avalia de forma prática, levantando questões que ocorrem durante a execução de processos em um mundo real.

Na próxima seção é apresentado o problema de forma consistente e nas suas subseções é demonstrada a solução a ser aborda e como será construído o ambiente de simulação. O cronograma de atividades mostras o tempo dedicado a cada uma das atividades que foram definidas na seção anterior e por fim a conclusão resume o que é esperado com este trabalho.

2. Conceituação

Um sistema distribuído de bases de dados, não deixa de ser visto sobre alguns aspectos como um SGBD local, ou seja, este deve respeitar a principal propriedade que define um SGBD, conhecida como ACID (Atomicidade, Consistência, Isolamento e Durabilidade)[4], só que de uma forma global.

Em um sistema local esta propriedade já apresenta uma certa dificuldade em ser respeitada, para tanto ao longo dos anos vários modelos foram apresentados para a sua implementação. Um que ganhou destaque ao longo dos últimos dos anos e passou a ser utilizado pelos principais fornecedores de SGDBs do mundo, é conhecido como *Snapshot Isolation*. *Snapshot Isolation* garante que uma transação possa ver uma imagem consistente do banco de dados e que toda a tarefa de escrita mantém o banco em um estado consistente (respeito à regra Consistência)[6].

De forma simplificada, uma transação T ao dar início, realiza uma busca a todos valores que esta irá ler e/ou escrever no sistema, gerando assim uma fotografia dos dados naquele momento (não há a necessidade de manter bloqueios de leitura e escrita no sistema). Os dados que T manipula são os presentes nesta imagem, assim caso algo ocorra errado ao longo de sua execução a imagem é apenas descartada. Quando T desejar efetivar uma escrita a um dado D, é necessária a validação para verificar se T está em concorrência com alguma outra Tn, note que o único problema presente é se a Tn efetivou uma escrita em D depois que T leu, assim o valor de D utilizado por T está desatualizado e cancelamento da transação ocorre.

Há outras particularidades sobre o funciomanento deste algoritmo e podem ser encontrados na literatura, o objetivo aqui não é a compreensão completa deste e sim somente de uma parte. A próxima seção apresenta o algoritmo que é empregado neste trabalho.

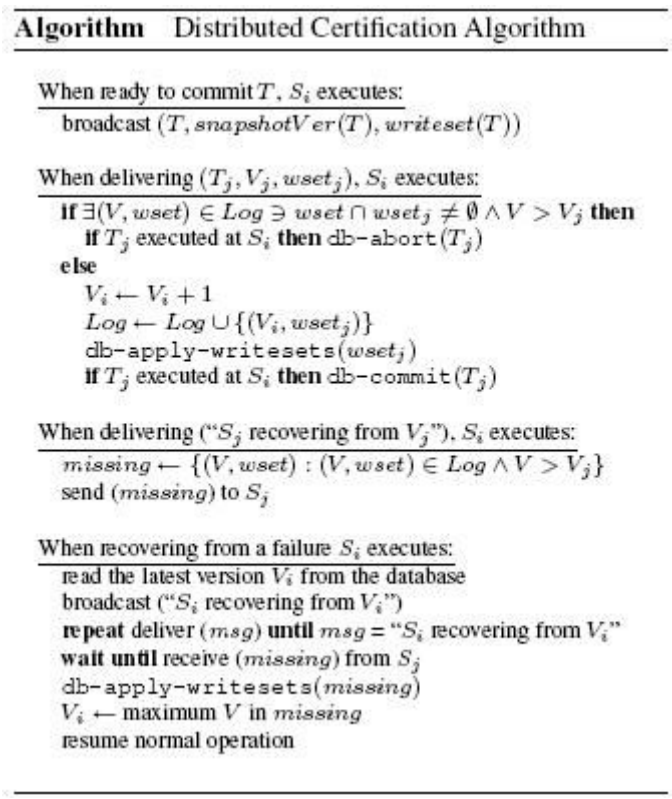

Figura 1 Algoritmo de Certificação Distribuída

2.1. Modelo Proposto

A partir do funcionamento do mecanismo de *Snapshot Isolation* os autores [1] realizaram um estudo de caso sobre as variações possíveis na construção de uma alternativa ao algoritmo que permitisse o seu uso em bancos de dados distribuídos. Este estudo levou ao final do trabalho a construção de dois procedimentos. O primeiro utiliza um processo certificador global para o sistema, onde todas as transações distribuídas enviam os snapshots e consultam sobre os demais, garantindo assim o isolamento de todas as transações. O modelo com um certificador global limita o uso de um sistema distribuído, pois impõe a todo o sistema a necessidade do autenticador global, limitando assim a autonomia e impondo a possibilidade de falha do certificador, assim este modelo será desconsiderado. O segundo modelo proposto é uma variação do primeiro, mas neste todos os bancos distribuídos funcionam como unidades certificadoras dos diversos *snapshots*.

A figura 1 mostra o algoritmo e o seu funcionamento ocorre da seguinte forma. Toda transação T é processada de forma isolada e local, até o momento que esta deseja realizar uma efetivação das modificações realizadas (*commit*). Quando chega este momento é enviado um *brodcast* para todos os bancos distribuídos contendo o identificador de T, o *snapshotVer*(T) que este estava utilizando e o conjunto de escrita *writeset*(T). O algoritmo verifica se a efetivação de T entra em conflito com alguma outra Tn que efetivou antes de T, caso verdade T é abortada senão T é validada e efetivada no banco e local e por seguinte em todas as demais bases. Este modelo proposto opera sobre a seguinte condição,

o sistema distribuído opera sobre a troca ordenada de mensagens. Devido a esta ultima consideração é possível observar que o processo de funcionamento do algoritmo não implica em um cancelamento global do sistema ou em cascata. Ou seja, antes de validar uma escrita T envia para todo o sistema a sua intenção de alterar aquele dado em especial. Como o funcionamento do sistema se baseia na troca de mensagem ordenada, então T só valida localmente a possibilidade de conflito em D, já q se uma outra transação rodando em outro servidor também estiver tentando escrever em D, esta intenção estará assinalada no vetor de escritas. Mas este sistema não considera um possível atraso na entrega da mensagem, assim não fica claro o que ocorre com D caso duas transações Ti e Tj em servidores S1 e S2 respectivamente, diferentes informam sobre a intenção de alterar D ao mesmo tempo. Afinal, S1 não possuem a informação de que T2 de S2 está querendo escrever em D e o caso contrário também é verdade.

2.2. Simulação

O que foi proposto demonstra-se de forma interessante e plausível, tirando algumas questões que poderão ser respondidas de forma prática e assim demonstrar se o algoritmo proposto atende aos requisitos para garantir a consistência global do sistema de dados distribuído. O trabalho [1] faz a proposta e apresenta dados gerados por um brenchmark comercial, analisando os custos de performance do sistema, ficando questões não respondidas sobre o seu comportamento funcional.

Este trabalho tem o intuito de provar de forma prática o que está conceituado no algoritmo. Desta forma a presente proposta tem a intenção de responder perguntas do estilo: o que ocorre ao sistema quando um determinado tipo de falha ocorre. Este não tem por intuito validar este algoritmo como sendo o melhor em termos de desempenho ou como o mais correto, uma outra série de algoritmos já foram propostos e a maior parte dos SGDBs atuais de grande porte, já trabalham com distribuição de dados. A única intenção aqui é demonstrar o comportamento do sistema sobre simulações da realidade

O trabalho será construído da seguinte forma, a ferramenta Spread[5] dará o suporte para simular bancos de dados distribuídos que realizam a troca de mensagens de forma ordenada entre todas os nodos do sistema, garantindo assim a base de funcionamento do sistema.

Já o algoritmo será implementado na linguagem Java [2], onde será simulado um *snapshot* básico e simples. O objetivo desta base de testes é ir aumentando o grau de complexidade do sistema simulado para que este demonstre ao máximo o funcionamento do algoritmo em um sistema real. Note que a comparação com a realidade é um tanto quanto subjetiva, assim fica proposto aqui demonstrar como o sistema se comporta em uma rede com N servidores, como é o funcionamento deste quando N-2 servidores falham.

3. Cronograma de atividades

A tabela 1 apresenta o cronograma de desenvolvimento deste projeto. Levando-se em consideração a necessidade de suprir de imediato a falha de conhecimento na ferramenta que irá servir como camada base para a simulação dos sistemas distribuídos.

As atividades que concentraram mais tempo para realização serão a de construção do simulador e a definição dos possíveis erros a serem tratados pela ferramenta, pois diante da grande quantidade de possibilidades, é necessário definir quais são os mais pertinentes a estudo de caso.

Ao final será construído um relatório demonstrando as variáveis encontradas e o comportamento do sistema simulado e suas variações.

	Abril -2 quinzena	Maio – 1 quinzena	Maio – 2 quinzena	Julho – 1 quinzena
Estudo do Spread	X			
Construção do simulador		X		
Definição dos problemas		X	X	
Analise dos resultados			X	X
Entrega do resultado				X

4. conclusão

A presente proposta tem por objetivo discutir assuntos que primeiramente não foram tratados no artigo, como é o funcionamento real do algoritmo de replicação de bases de dados utilizando *snapshot isolation.*

Esta avaliação será de extrema importância, pois ajuda a compreender em como o processo de troca de mensagens ordenadas, juntamente com um procedimento para controle de imagens de dados, em um SGDB podem lhe dar a característica de sistema de bancos de dados distribuídos, com consistência global.

As questões que devem ser respondidas aqui são meramente especulativas e nem de uma forma mais completa este trabalho seria capaz de apresentar todas as possíveis variações de comportamento que um sistema distribuído pode apresentar.

Referências

[1] Elnikety, S., Pedone, F., Zwaenepoel, W., (2005) Database Replication Using Generalized Snapshot Isolation in *Symposium Reliable Distributed Systems* SRDS

[2] Java, http://java.sun.com/ , abril 2007

[3] Pedroni, L. M., Ribeiro, H.G, (2006) Replicação em Banco de Dados PostgreSQL in Escola Regional de Bancos de Dados – ERDB, Caxias do Sul

[4] Silberchatz, A.,Korth, H. F., Sudarshan, S. (1999) , Sistema de Banco de Dados , pg. 557, Makron Books do Brasil.

[5] The Spread Toolkit disponível em: http://www.spread.org/ , abril 2007

[6] Wikipedia, The free encyclopedia, abril 2007