

Memórias Transacionais

Luciano Jerez Chaves

RA: 079759

lchaves@ic.unicamp.br



Instituto de Computação
Universidade Estadual de Campinas



23 de Junho de 2010

Contextualização

Evolução dos sistemas computacionais

- Computadores com apenas um processador
- Aumento da frequência de operação
- Lei de Moore

Limitações

- Limites físicos
- Dissipação de calor

Computadores *multicores*

Contextualização

Programação paralela

- Extrair paralelismo das aplicações
- Dificuldade em projetar e implementar sistemas paralelos

Controle de concorrência

- Compartilhamento de memória → região crítica
- Soluções envolvem *locks*, semáforos, etc.
- Serialização da execução

Transações

Semântica de programação onde a computação é executada como se fosse a única existente

Propriedades ACID

- Atomicidade
- Consistência
- Isolamento
- Durabilidade

Definição sintática

```
atomic {  
    if (x != NULL)  
        x.function();  
    y = true;  
}
```

Memórias transacionais

Implementação responsável por executar transações garantindo a corretude nas operações

Características de implementação

- Verificação de conflitos

Eager monitora em tempo real

Lazy verifica antes do *commit*

- Resolução de conflitos

Eager *snapshot* antes do início

Lazy alterações em um *buffer*

Memórias transacionais

- Isolamento entre transações
 - Forte** somente acessos dentro das transações
 - Fraco** todos os acessos
- Gerenciamento de contenção
 - Committer wins** na etapa de *commit*, a transação que identifica a colisão conclui
 - Requester wins** durante um acesso concorrente, a transação que faz o acesso por último conclui
 - Committer stalls** durante um acesso concorrente, a primeira transação que fez o acesso conclui

Modelos de Implementação

Memória Transacional em *Hardware* (HTM)

- Primeiras abordagens propostas
- Maximizar desempenho
- Eliminar suporte de compiladores

Memória Transacional em *Software* (STM)

- Maior flexibilidade de implementação
- Menor custo de produção
- Menos limitações intrínsecas

Memória Transacional Híbrida

- Combinação dos anteriores
- Sistemas STM apoiados por *hardware*
- Dificuldade de implementação × Ganho de desempenho

Análise qualitativa

Experimento realizado por Pankatius *et al* em 2009

- TM reduz o esforço de programação
- TM reduz o tempo gasto em depuração
- Pode ser utilizado em conjunto com técnicas tradicionais
- Ajuste de desempenho complexo (indeterminismo)