

Proposta de Rápida Recuperação por Redução de Log

Luiz Gustavo C. Xavier¹, Fernando Luís Dotti²,
Cristina Meinhardt¹, Odorico M. Mendizabal¹

¹Instituto de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)

²Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

l.gustavo.x@posgrad.ufsc.br, fernando.dotti@pucrs.br,
{odorico.mendizabal, cristina.meinhardt}@ufsc.br

Resumo. *Logs são cruciais no desenvolvimento de aplicações distribuídas tolerantes a falhas. Este trabalho apresenta uma abordagem para acelerar a recuperação em protocolos de recuperação baseados em log. A abordagem explora a eliminação consciente de comandos de maneira pouco intrusiva à aplicação, resultando em arquivos de logs menores sem ferir a consistência de estado da aplicação, permitindo uma rápida recuperação.*

1. Introdução

Mecanismos de *log* desempenham um papel central no desenvolvimento de sistemas de gerenciamento de banco de dados [Mohan et al. 1992], replicação e protocolos de coordenação e integração de dados [Kreps et al. 2011]. Porém, além de adicionar custos durante a operação normal, o processamento de um *log* de comandos afeta diretamente o tempo de recuperação. Mecanismos de *log* tradicionais (SL) normalmente não se beneficiam da semântica de comandos da aplicação, sendo desconhecido para o protocolo de recuperação se um resultado de comando é substituído posteriormente ou se sua execução não modifica o estado da aplicação em recuperação.

Neste trabalho, é apresentada uma abordagem para acelerar a recuperação em protocolos baseados em *log*. Na abordagem de *log* proposta (PL) os comandos são analisados em lotes, e a redução é realizada considerando somente os comandos contidos no lote. Uma vez que PL se baseia na semântica da aplicação para o processo de redução, é assumido o modelo de dados de um armazenamento de valor-chave executando comandos de leitura e escrita de uma única variável. Nesse caso, as operações de leitura e escritas subsequentes são exemplos de comandos desnecessários, onde apenas o último comando de escrita de uma variável por lote precisa ser mantido para atingir um estado consistente.

2. Avaliação Experimental e Considerações Finais

Para avaliar a rapidez obtida com a execução do protocolo de recuperação apresentado, são analisados os arquivos de *log* gerados em diferentes configurações de intervalo em relação à abordagem tradicional, comparando o número de comandos e o tamanho total dos arquivos. Os resultados apresentados foram obtidos com a execução de um protótipo de aplicação de armazenamento chave-valor desenvolvida em Go, com a geração de carga de trabalho usando o *Yahoo! Cloud Serving Benchmark* (YCSB).

Na Figura 1a é ilustrada a redução obtida no número de comandos para cada carga de trabalho, onde o tamanho do lote é configurado em 1, 10, 100 e 1000 comandos. Para todas as cargas de trabalho, a abordagem SL registrou todos os 10^6 comandos no *log*. O mesmo pode ser dito para o cenário de 100% escritas YCSB-AW, apresentando o mesmo número de comandos que SL independente do tamanho do lote. Isso é explicado pela carga de trabalho de somente escritas com chaves de acesso distribuídos uniformemente. Em um cenário onde há grande incidência de escritas subsequentes sobre uma mesma variável (YCSB-AWL), é possível perceber que quanto maior o tamanho do lote, maior a chance de tais operações serem identificadas e descartadas durante os procedimentos de *log*. Nesta mesma carga de trabalho, observa-se um ganho de 32% menos de comandos no PL1000. Todos os comandos são eliminados para YCSB-C, uma carga de 100% leituras, e obtém-se mais de 90% de redução em YCSB-B e YCSB-D.

A Figura 1b mostra as reduções de tamanho total do *log* em comparação com SL. Nos cenários PL1, a abordagem proposta apresenta uma penalidade de $\approx 17\%$ no tamanho total do *log* para cargas de trabalho YCSB-AW e YCSB-AWL e um aumento de 5% para YCSB-A. Isso é explicado pelo fato de que o PL gera um novo arquivo de *log* com seus metadados em cada redução de lote, e a soma de todos esses arquivos gerados possui um tamanho maior quando comparado a um único arquivo gerado em SL. Esse efeito não é significativo para outras cargas de trabalho, em que reduções de tamanho são mostradas para cargas de trabalho com leitura intensiva. Para tamanhos de lote ≥ 10 comandos, YCSB-A mantém uma redução de $\approx 20\%$. YCSB-AWL mostra benefícios incrementais em lotes maiores, com uma redução de 30% nos *logs* para PL1000. Diferenças maiores podem ser vistas para YCSB-C, B e D, onde uma diminuição de quase 100% é mostrada para o primeiro e $\approx 80\%$ para os demais em tamanhos de lote de 1000 comandos.

Este resumo apresentou uma abordagem de *log* que explora a semântica da aplicação para eliminar comandos desnecessários, provendo indícios de uma recuperação mais rápida. Trabalhos futuros irão avaliar o impacto real desta técnica na recuperação.

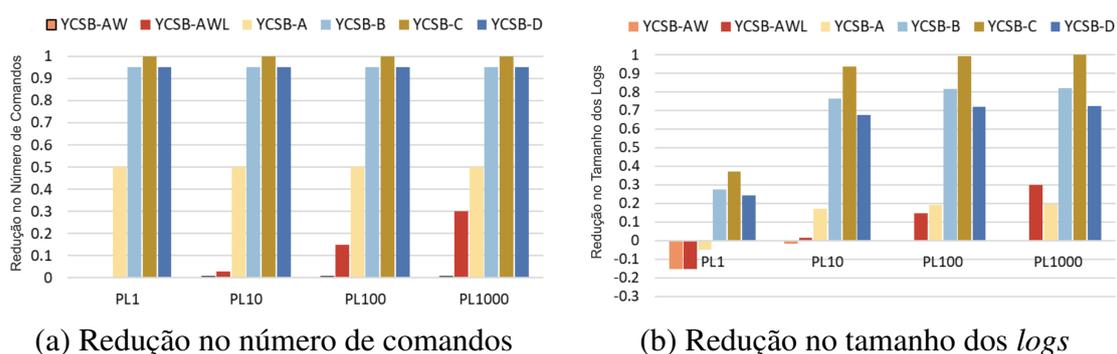


Figura 1. Avaliação dos arquivos de log gerados. Dados normalizados para SL.

Referências

- Kreps, J., Narkhede, N., Rao, J., et al. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, volume 11, pages 1–7.
- Mohan, C., Haderle, D., Lindsay, B., Pirahesh, H., and Schwarz, P. (1992). Aries: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems (TODS)*, 17(1):94–162.