Avaliação do mecanismo de checkpoint no HDFS em um cenário com falha de DataNode

Paulo V. M. Cardoso, Patrícia Pitthan Barcelos

Pós-Graduação em Ciência da Computação (PGCC)
Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brazil

pcardoso@inf.ufsm.br, pitthan@inf.ufsm.br

Resumo. *A técnica de Checkpoint and Recovery apresenta-se de forma eficiente no contexto de tolerância a falhas, atenuando problemas de confiabilidade e disponibilidade em sistemas de alto desempenho. O Apache Hadoop, usado para trabalhar com quantidades massivas de dados, implementa checkpoint estático em seu sistema de arquivos distribuído. Este trabalho apresenta uma validação do checkpoint no Hadoop através da indução de falhas no DataNode.*

1. Introdução

O contexto da computação de alto desempenho exige o uso de um número cada vez maior de componentes em sistemas computacionais. Essa característica influencia a queda de confiabilidade e disponibilidade, já que tais sistemas são mais propensos a falhas. Uma técnica de tolerância a falhas bastante utilizada é o *Checkpoint and Recovery* (CR), que consiste no salvamento do estado do serviço (*checkpoint*) e na recuperação pós falha.

O Apache Hadoop, *framework* usado para processar e armazenar grandes quantidades de dados, usa a técnica de CR. Porém, o *checkpoint* no Hadoop possui atributos de configuração estáticos. Ou seja, o período de salvamento de *checkpoints* não pode ser alterado em tempo de execução, sendo que a escolha do atributo é determinante tanto para o desempenho de aplicações quanto para o nível de confiabilidade do sistema.

O trabalho apresenta uma validação do mecanismo de *checkpoint* estático do Hadoop em situações de falhas no DataNode, elemento responsável pelo armazenamento de dados do HDFS. As falhas são induzidas em períodos específicos de execução, de forma que o DataNode não volte a ser executado (falha de *crash*). Assim, o sistema deve manter sua execução e realocar os dados perdidos para novos nós através dos *checkpoints* salvos. A validação é feita por uma análise de desempenho do tempo de execução do Hadoop.

2. Checkpoint

O *checkpoint* no Hadoop é implementado para tolerar falhas no HDFS [White 2015]. Para isso, o namespace do HDFS é replicado no arquivo FSImage, armazenado em disco no sistema de arquivos local do NameNode. Esse arquivo mantém informações sobre o mapeamento de blocos para arquivos e propriedades do sistema. Para evitar a criação de novo FSImage a cada operação do HDFS, um *log* de edições (EditLog), também mantido em disco local, armazena as últimas transações realizadas após a criação do FSImage.

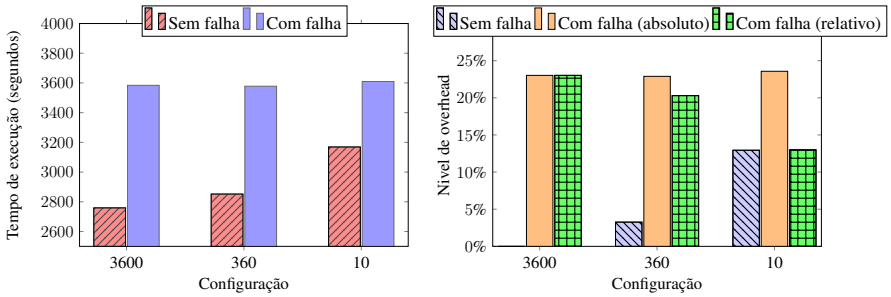
O *merge* entre o FSImage e o EditLog consiste no processo de *checkpoint*. Esse procedimento é realizado quando o NameNode inicia e, posteriormente, é feito de forma

periódica. O intervalo de *checkpoint* padrão do Hadoop é 3600s, mas pode ser disparado se o HDFS atingir um número determinado de transações. O *checkpoint* é realizado pelo SecondaryNameNode (SNN), que mantém uma cópia do FSImage e a atualiza a cada *checkpoint*, solicitando o arquivo de edições ao NameNode.

3. Resultados e Trabalhos Futuros

A experimentação foi realizada a partir do *benchmark* TestDFSIO, disponibilizado pela distribuição do Hadoop. A aplicação foi usada para testar o HDFS a partir de operações de escrita com 20 arquivos de 16GB cada. As execuções foram executadas com 20 amostras em 8 nós da plataforma Grid5000 [Grid'5000 2017].

A falha de *crash* é induzida no DataNode do nó executor no tempo de 20% da média *baseline* (3600 segundos de *checkpoint*) com o comando $kill$ do Linux, emulando-se falha de *crash*. Utilizou-se o fator de replicação 3 (padrão). A Figura 1(a) mostra os resultados com variação do período de *checkpoint*. O tempo médio de execução é mais alto com frequências maiores. Com falhas, porém, a sobrecarga tem um comportamento linear. *Checkpoints* mais atualizados tendem a economizar o tempo de atualização do *namespace* com EditLogs menores, já que a recuperação de réplicas perdidas requisita uma quantidade considerável de operações no NN, a fim de manter o fator de replicação.



(a) Tempo de execução (b) Análise de sobrecarga.

Figura 1. *Overhead* da variação de *checkpoint* no TestDFSIO.

A Figura 1(b) mostra o *overhead* observado em um cenário sem falhas, relacionado ao *baseline* (absoluto) e ao resultado de cada variação no teste sem falha (relativo). O impacto do período de *checkpoint* não interfere de forma significativa no *overhead* absoluto da aplicação, porém a observação relativa revela que uma periodicidade maior indica uma recuperação mais eficiente. Percebe-se que a exigência por um nível de confiabilidade mais alto, de fato, auxilia no processo de recuperação. Por outro lado, essa configuração sofre com sobrecargas em cenários sem falhas.

Trabalhos futuros investigarão o comportamento das aplicações em cenários de falha transiente no NameNode, já que o *checkpoint* é um elemento essencial de recuperação do nó mestre do HDFS. A partir dessas análises, o uso do *checkpoint* dinâmico proposto será aplicado de forma a verificar sua usabilidade no contexto de falhas do Hadoop.

Referências

Grid'5000 (2017). *Grid5000 Homepage*. http://www.grid5000.fr/, (acessado em dezembro de 2017).

White, T. (2015). *Hadoop: The Definitive Guide, 4th Edition*. "O'Reilly Media, Inc.".