

# Proposta de validação de uma arquitetura para checkpoint dinâmico no Apache Hadoop

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 análise da eficiência de recuperação de um sistema em determinados cenários de falha é essencial para validar a viabilidade dos mecanismos de tolerância a falhas. Este trabalho propõe uma alternativa dinâmica para o estabelecimento de checkpoints no Apache Hadoop e define um cenário para sua validação com falhas no nó mestre da ferramenta.*

## 1. Introdução

Sistemas de alto desempenho devem evitar a ocorrência de erros, tornando-se necessário o emprego de técnicas de tolerância a falhas. O *Checkpoint and Recovery* (CR) consiste em uma técnica reativa de recuperação de erros por retorno cujo objetivo é conduzir o sistema a uma condição consistente, através do salvamento periódico do seu estado em momentos anteriores a falhas (*checkpoint*). A recuperação acontece após uma falha e se resume em recuperar o andamento normal do sistema.

No *framework* Apache Hadoop [White 2015], projetado para armazenar e processar grandes quantidades de dados, o *checkpoint* é usado pelo seu sistema de arquivos distribuído (HDFS - *Hadoop Distributed File System*) para alcançar tolerância a falhas. Neste sentido, o *checkpoint* é importante na manutenção do NameNode, já que este é um elemento essencial no que se refere à disponibilidade.

Contudo, o *checkpoint* no Hadoop possui configuração estática, pois não permite que o período entre salvamentos seja alterado sem que os serviços do *framework* sejam interrompidos. A partir dessa observação, este trabalho apresenta um mecanismo de *checkpoint* dinâmico para o Hadoop, a fim de garantir que o mesmo possa adaptar-se ao contexto de uso dos seus recursos. Para a validação, é definido um cenário de falhas transientes no NameNode a partir de modificações no gerenciador de aplicações do Hadoop.

## 2. Checkpoint Dinâmico

o Hadoop possui atributos de configuração de *checkpoint* estáticos, que são definidos antes da execução do *framework*. Por isso, a confiabilidade do sistema depende, dentre outros fatores, da definição de um período ideal entre *checkpoints*. Nesse caso, *checkpoints* mais frequentes aumentam o nível de confiabilidade mas comprometem o desempenho, enquanto *checkpoints* menos frequentes elevam o tempo de recuperação.

O mecanismo de *checkpoint* dinâmico, proposto em [Cardoso and Barcelos 2015], tem como objetivo tornar a configuração do CR no Hadoop adaptável para tornar este processo menos intrusivo, visando não comprometer a confiabilidade e o desempenho do

HDFS. A partir de eventuais mudanças na periodicidade dos *checkpoints*, o mecanismo dinâmico pode oferecer soluções adequadas de acordo com o contexto de uso do sistema.

A arquitetura proposta é constituída por alterações no HDFS, monitores independentes, além de um coordenador. Em cada elemento que seja necessário acompanhamento, um processo monitor é iniciado para identificar alterações no comportamento do nó (trocas de contexto). A partir de métricas de desempenho, o monitor envia uma mensagem ao supervisor – módulo central de monitoramento –, que indica um novo período de *checkpoint*. O coordenador do mecanismo, implementado com o auxílio do *framework* Zookeeper, armazena o novo atributo e invoca um alerta de mudança ao HDFS.

O objetivo é garantir um intervalo ideal para cada situação identificada, para que o *checkpoint* do Hadoop esteja sempre adaptado ao ambiente. Isto é, de acordo com o nível de utilização dos recursos e das métricas definidas, o atributo é atualizado e alterado sem a necessidade de interromper qualquer serviço da ferramenta. Com um intervalo adaptado em tempo de execução, o ambiente pode ser aproveitado de forma mais eficiente.

### 3. Validação

A validação do mecanismo define um cenário geral de falhas transientes no NameNode para analisar a eficácia da recuperação do Hadoop com os mecanismos estático e dinâmico para o *checkpoint*. A falha transiente será induzida em momentos específicos da execução de aplicações, com o comando *kill* do Linux. O NameNode deve ser reiniciado instantes depois da falha para que se faça uma análise de sua recuperação.

Como uma falha no NameNode é caracterizada como um ponto único de falha (SPOF), os serviços que dependem do nó mestre são finalizados com erro. Assim, qualquer aplicação que use o HDFS sofre essa restrição. Para que uma falha transiente possa ser emulada, a validação propõe um mecanismo de espera no ApplicationMaster (AM) de cada aplicação. O AM é responsável pela negociação de recursos para determinada aplicação, além da coordenação dos *containers* de execução do Hadoop.

Para que os *containers* não sejam afetados pela falha no NameNode, as modificações propostas no AM incluem um *timeout* máximo de espera por uma possível recuperação do NameNode, além do tratamento da etapa de recuperação. Desta forma, a dinamicidade do mecanismo proposto pode ser validada perante um cenário de falhas no elemento mestre do HDFS, definido como meta de tolerância a falhas do *checkpoint*.

### 4. Próximos Passos

Com a definição dos cenários de falha, serão executados testes de desempenho com objetivo de analisar o tempo de recuperação do HDFS. Além da validação do mecanismo dinâmico nos cenários definidos, será feita uma comparação com o mecanismo de *High Availability* do Hadoop, que define uma arquitetura para eliminar o SPOF (*Single Point of Failure*) do NameNode.

### Referências

- Cardoso, P. V. and Barcelos, P. P. (2015). Avaliação de uma arquitetura de configuração dinâmica para o checkpoint no hdfs. *XII Encontro Anual de Tecnologia da Informação*.
- White, T. (2015). *Hadoop: The Definitive Guide, 4th Edition*. "O'Reilly Media, Inc."