

# Escalonamento Adaptativo para o Apache Hadoop

Guilherme Weigert Cassales<sup>1</sup>

Andrea Schwertner Charão<sup>1</sup>

{cassales, andrea}@inf.ufsm.br

<sup>1</sup>Departamento de Linguagens e Sistemas de Computação

Centro de Tecnologia

Universidade Federal de Santa Maria

18 de setembro de 2015

# Roteiro

Introdução

Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

# Roteiro

Introdução

Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

# Introdução

- ▶ *MapReduce*.
- ▶ *Framework* mais conhecido: Hadoop [1].
- ▶ Arquitetura utilizada: *cluster*.
- ▶ Principal objetivo: processar grandes volumes de dados, em geral textuais.

---

<sup>1</sup>Apache, Apache hadoop. <http://hadoop.apache.org/docs/r2.6.0/index.html>.

# Características

Características do Hadoop decorrentes do projeto:

- ▶ projetado para *clusters* **homogêneos**;
- ▶ configuração feita por arquivos de texto com valores *default*;
- ▶ arquitetura mestre e escravo;
- ▶ separa as atribuições em dois grandes módulos, HDFS (armazenamento) e YARN (processamento);
- ▶ **não possui capacidade de adaptação a ambientes levemente dinâmicos.**

# Objetivo

## Objetivo Principal

- ▶ Introduzir sensibilidade ao contexto nos mecanismos de escalonamento do Apache Hadoop através da coleta e transmissão de dados.

# Roteiro

Introdução

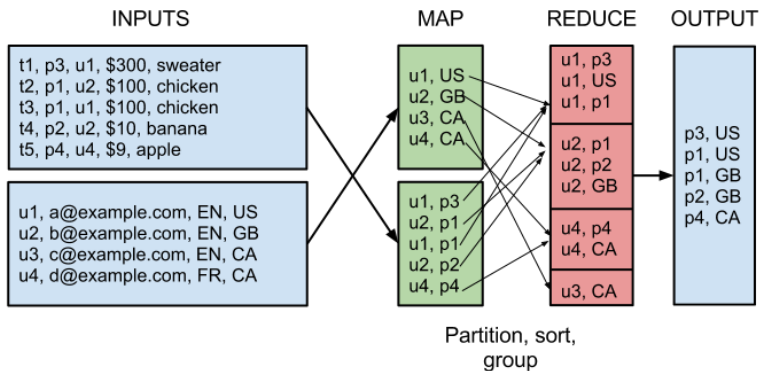
Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

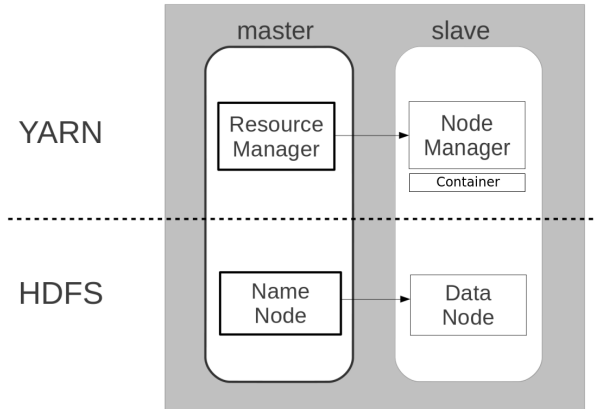
# MapReduce



<http://blog.matthewrathbone.com>



# Arquitetura do Hadoop



# Sensibilidade ao contexto

- ▶ Contexto
  - ▶ "qualquer **informação** que pode ser utilizada para **caracterizar** a **situação** de uma entidade (pessoa, lugar ou objeto) considerada **relevante** para a **interação** entre usuário e aplicação" (DEY, 2001).
- ▶ Sensibilidade ao Contexto
  - ▶ "se refere a habilidade de uma aplicação de **detectar e responder as mudanças no ambiente** de execução" (Maamar; Benslimane; Narendra, 2006).

# Trabalhos Relacionados

## Novos métodos de escalonamento

1. (Kumar et al., 2012).
2. (Tian et al., 2009).
3. (Rasooli and Down, 2012).
4. (Isard et al., 2009).
5. (Zaharia et al., 2008).
6. (Chen et al., 2010).

Classificação E/S ou CPU [1-3].

Grafo capacidade-demanda [4].

Heurística progresso da tarefa [5].

Dados históricos, escala própria [6].

## Melhoria na distribuição de recursos.

1. (Sandholm and Lai, 2010).
2. (Xie et al., 2010).

Taxa de gasto [1].

*Data locality* [2].

# Roteiro

Introdução

Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

# Etapas propostas

1. Inserção de um coletor de contexto.
2. Criação de uma comunicação entre os nós para melhor gestão dos recursos disponíveis.

# Coletor de contexto

- ▶ Desenvolvido para o projeto PER-MARE [4]
- ▶ Baseado na API padrão de monitoramento do Java [5]
- ▶ Novos coletores podem ser integrados para mais recursos.

---

<sup>4</sup> STIC-AmSud, PER-MARE project. <http://cosy.univ-reims.fr/PER-MARE>

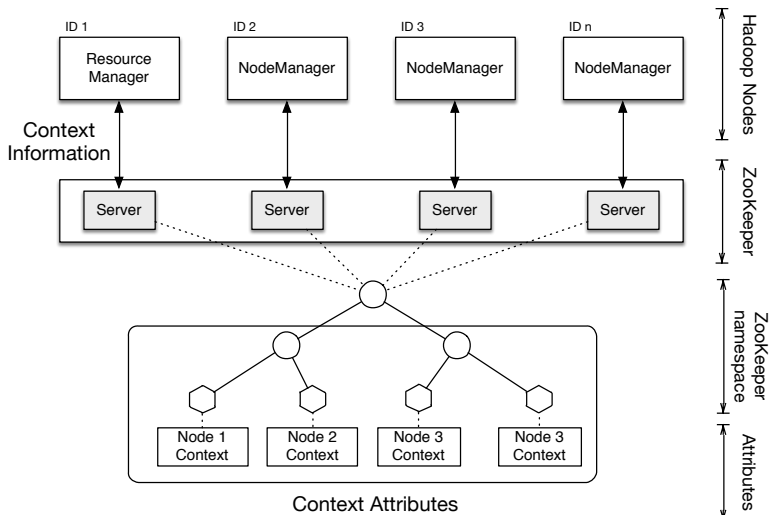
<sup>5</sup> Oracle, Overview of java se monitoring and management.

<http://docs.oracle.com/javase/7/docs/technotes/guides/management/overview.html>

# Comunicação entre nós

- ▶ Recursos de cada nó nunca são atualizados no escalonador.
- ▶ Problemas inerentes da comunicação distribuída.
- ▶ Solução: ZooKeeper.

# Comunicação entre nós





# Roteiro

Introdução

Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

# Configuração do ambiente

- ▶ Cluster genepi, Grid'5000.
- ▶ 1 mestre, 4 escravos;
- ▶ Cada nó possuindo: 2 CPUs Intel(R) Xeon(R) E5420 2.50 GHz (totalizando 8 cores por nó) e 8 GB de memória RAM;
- ▶ Sistema operacional Ubuntu-x64-12.04, JDK 1.8 e a versão 2.6.0 do Hadoop.
- ▶ Aplicação: TeraSort em um conjunto de dados de 15GB.

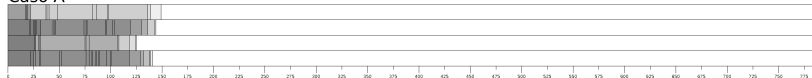
# Configuração dos experimentos

Quatro casos:

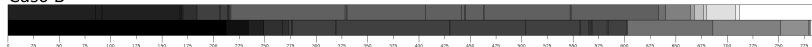
- A) Caso perfeito, recursos informados 100% e recursos disponíveis 100%.
- B) Comportamento *default* num ambiente dinâmico, recursos informados 100% e recursos disponíveis 50%.
- C) Atualização ocorre antes do início do *job*. Recursos disponíveis 50% e recursos informados 50%.
- D) Atualização ocorre após o início do *job*. **Inicialização do job:** recursos disponíveis 50% e recursos informados 100%. **Durante o job:** recursos disponíveis 50% e recursos informados 50%.

# Resultados parciais

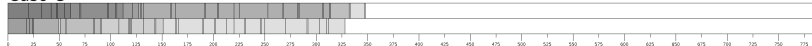
Caso A



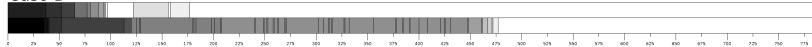
Caso B



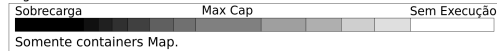
Caso C



Caso D



Legenda



# Roteiro

Introdução

Revisão de Literatura

Desenvolvimento

Experimentos e Resultados Parciais

Próximas Atividades

# Próximas Atividades

- ▶ Executar testes com aplicações de diferentes requisitos (CPU-bound, I/O-bound, etc) e diferentes domínios (*web search*, aprendizagem de máquina, etc) para confirmação da adaptabilidade da solução adotada.

# Escalonamento Adaptativo para o Apache Hadoop

Guilherme Weigert Cassales<sup>1</sup>

Andrea Schwertner Charão<sup>1</sup>

{cassales, andrea}@inf.ufsm.br

<sup>1</sup>Departamento de Linguagens e Sistemas de Computação

Centro de Tecnologia

Universidade Federal de Santa Maria

18 de setembro de 2015