Desenvolvimento de um Escalonador Sensível ao Contexto para o *Apache Hadoop*

Guilherme Weigert Cassales[1]
Orientadora: Profª Drª Andrea Schwertner Charão[1]

[1]Ciência da Computação
Universidade Federal de Santa Maria

20 de janeiro de 2014

informáticaufsm
Ciência da Computação

Roteiro

informáticaufsm
Ciência da Computação

Roteiro

informáticaufsm
Ciência da Computação

Introdução

- Relevância do *framework Apache Hadoop*.
- Problemas do *Hadoop*:
  - desenvolvimento focado para *clusters* homogêneos e dedicados;
  - dificuldades de adaptação a ambientes heterogêneos.
- Diversas soluções:
  - **escalonamento** de acordo com as *características* dos nós;
  - volatilidade dos nós utilizados no *cluster*;
  - **auto configuração** do ambiente de execução.

informáticaufsm
Ciência da Computação

Objetivo e Justificativas

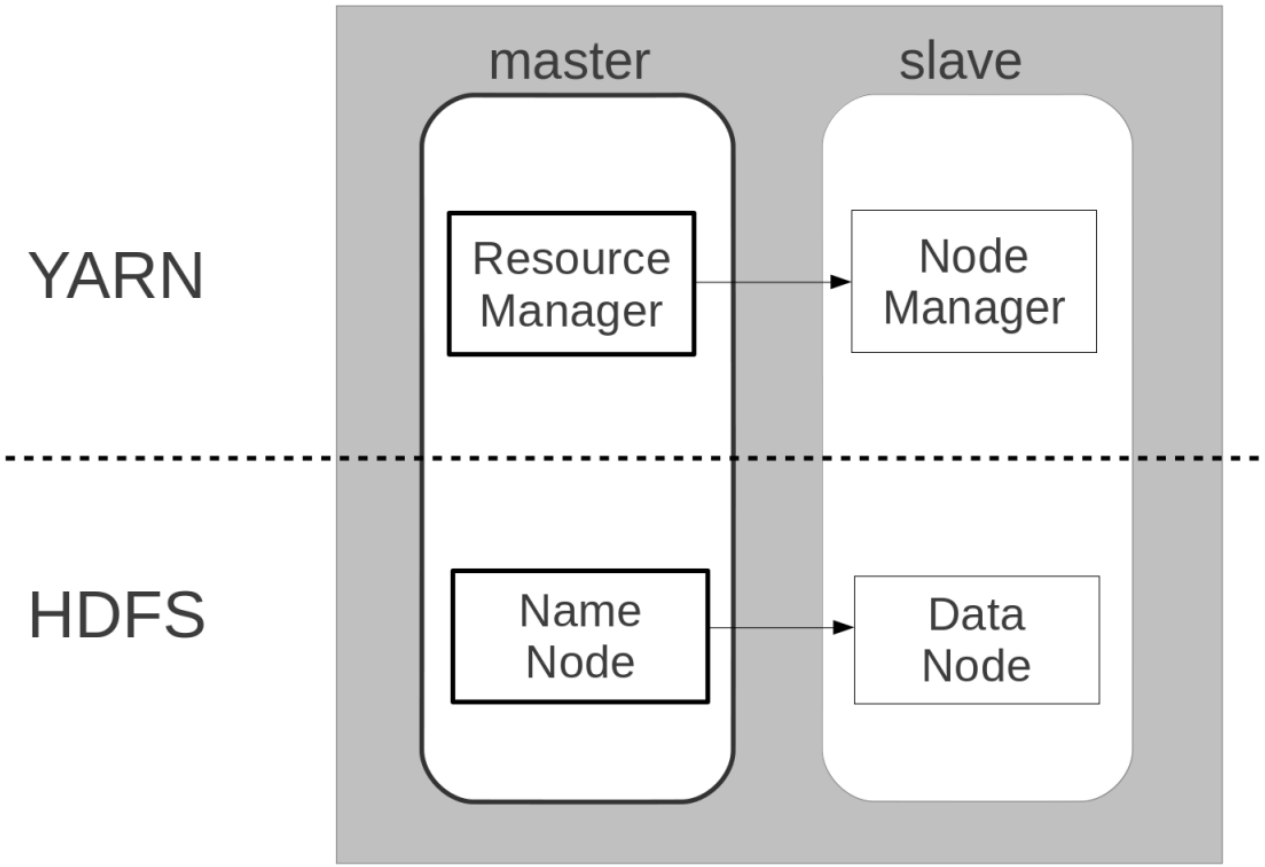
- Objetivo
  - Tornar o escalonamento do Apache Hadoop sensível ao contexto.
- Justificativas:
  - problemas de performance em situações de heterogeneidade;
  - problemas em manter um sistema homogêneo atualmente;
  - desperdício de capacidade (*green computing*).

informáticaufsm
Ciência da Computação

Roteiro

informáticaufsm
Ciência da Computação

Arquitetura Geral do *Hadoop*

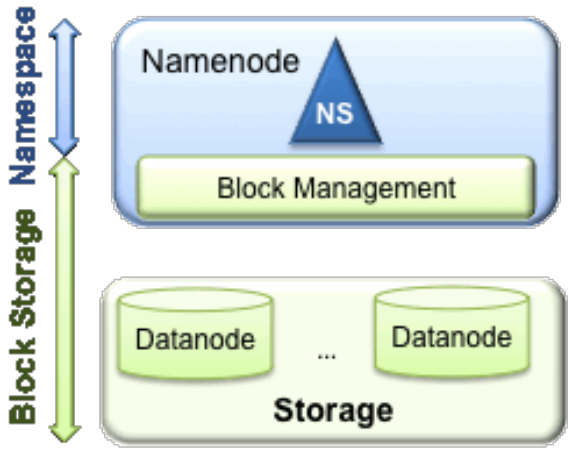


informáticaufsm
Ciência da Computação

Arquitetura Geral do *Hadoop*

HDFS

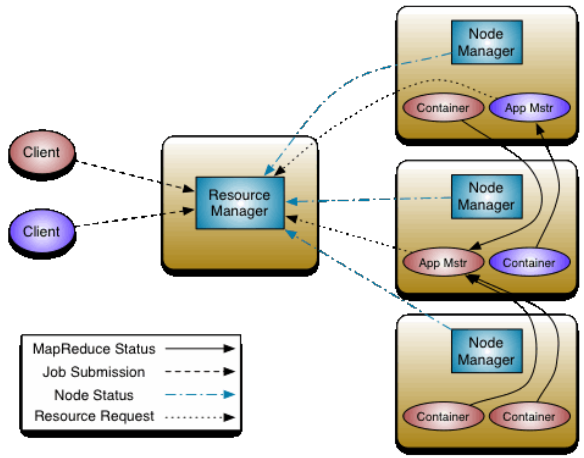
- *NameNode*
- *DataNode*



(HADOOP, 2013)

YARN

- *ResourceManager*
- *NodeManager*



(HADOOP, 2013)

informáticaufsm
Ciência da Computação

MapReduce

- O que é?
- Qual a relação do *MapReduce* com o YARN?
  - Classes;
  - *Containers*;
  - Requisição de recursos.
- Qual o ganho em utilizar?

informáticaufsm
Ciência da Computação

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).
- Como um *software* utiliza?
- Por que utilizar?

informáticaufsm
Ciência da Computação

Escalonadores Padrão

- *Internal*
  - FIFO.
- *Fair*
  - Distribuição igualitária de recursos.
- *Capacity*
  - Divisão de um *cluster* entre várias empresas.
  - Política de *MinShare*.
  - Estrutura para utilização recursos.
  - Configuração dos recursos precária.

Trabalhos relacionados

- (Kumar et al., 2012).
- (Rasooli; Down, 2012).
- (Chen et al., 2010).
- (Xie et al., 2010).
- (Tian et al., 2009).
- (Isard et al., 2009).
- (Zaharia et al., 2008).

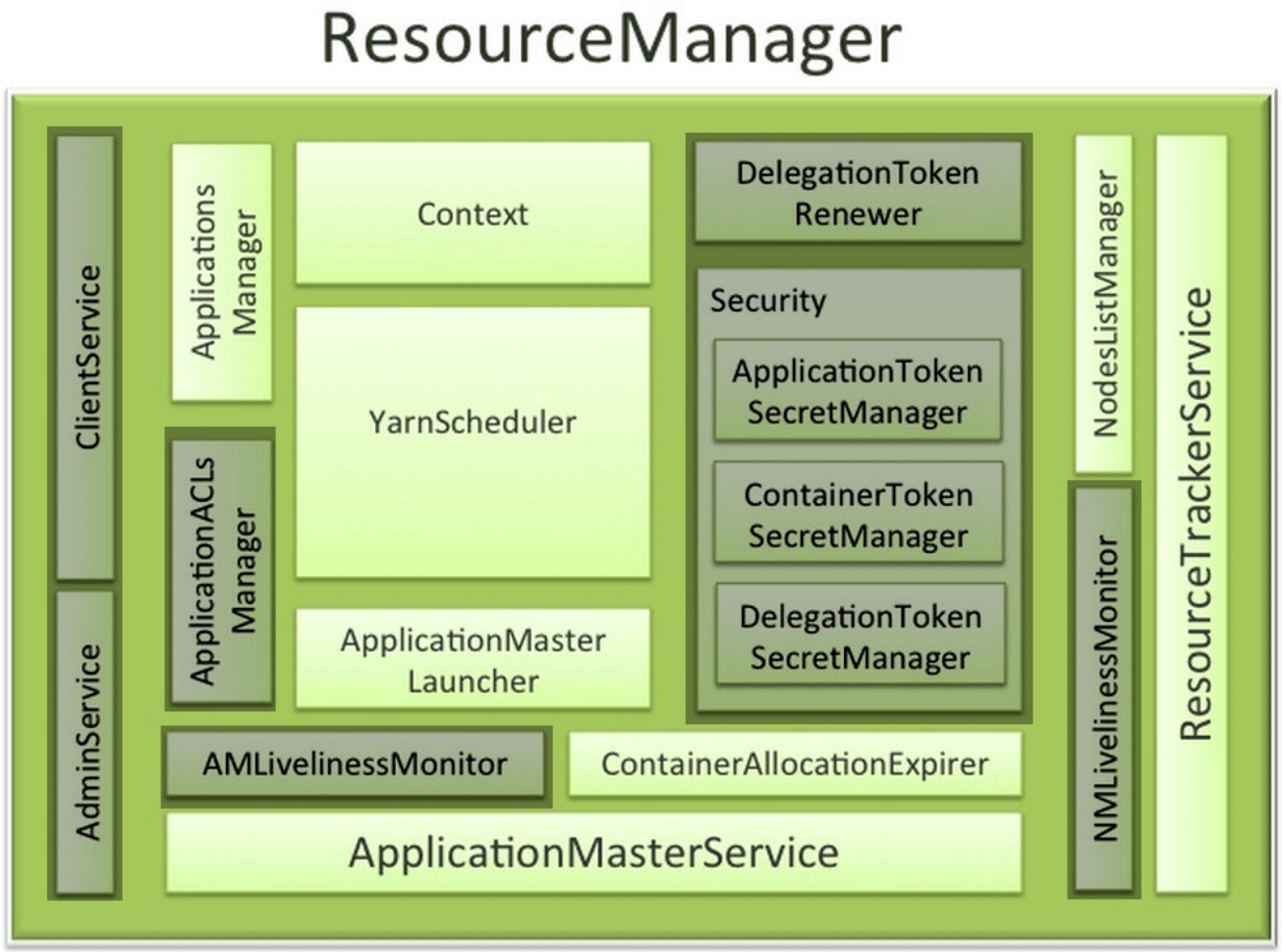
- Contextos mais comuns:
  - classificação de *jobs* e nós quanto ao potencial de E/S ou CPU;
  - avaliação do progresso da *task* na decisão de lançar ou não uma *task* especulativa.
- Objetivos mais comuns:
  - melhorar o *throughput*;
  - diminuir o tempo de resposta.

informáticaufsm
Ciência da Computação

Roteiro

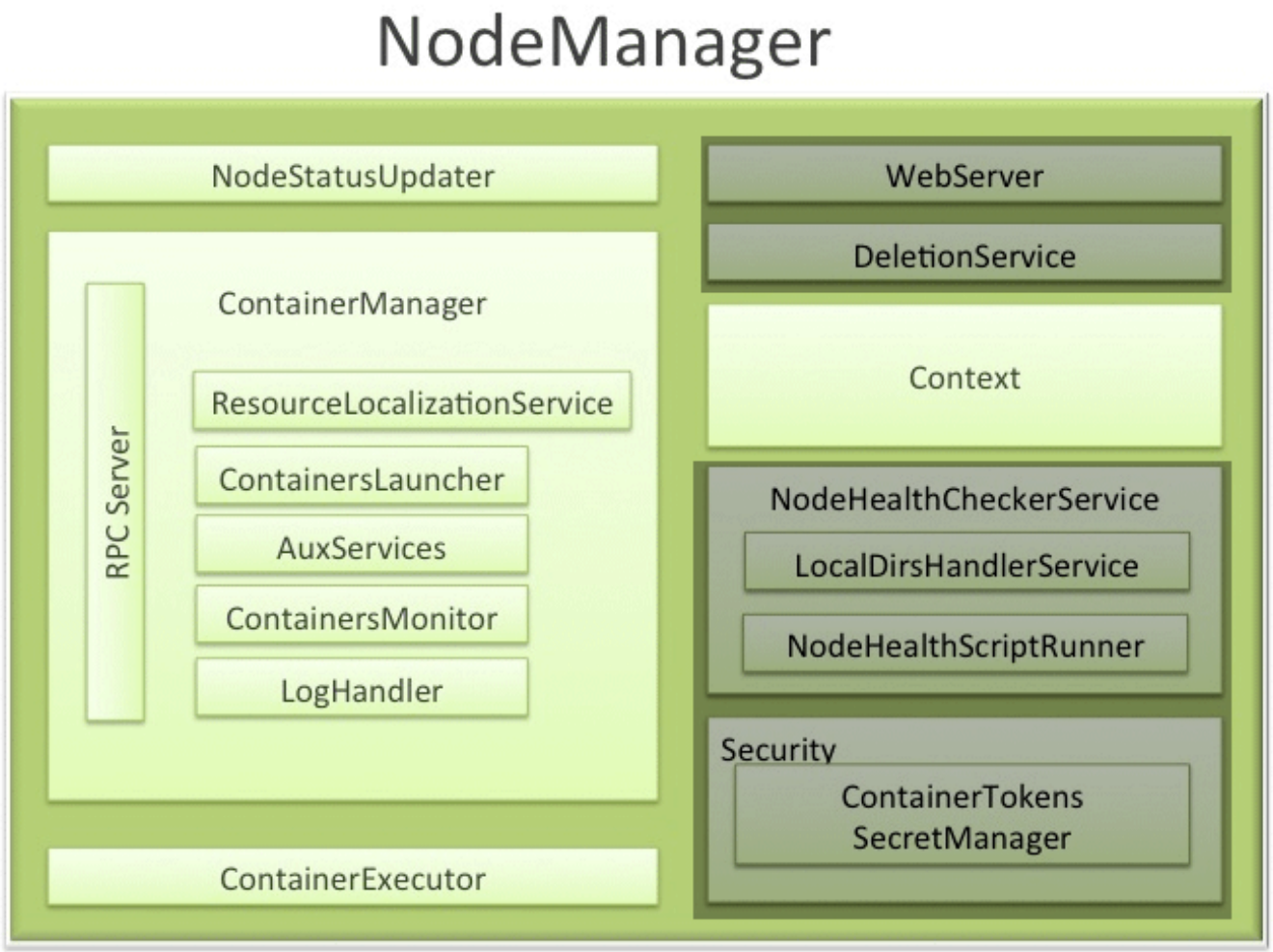
informáticaufsm
Ciência da Computação

Estudo da Arquitetura



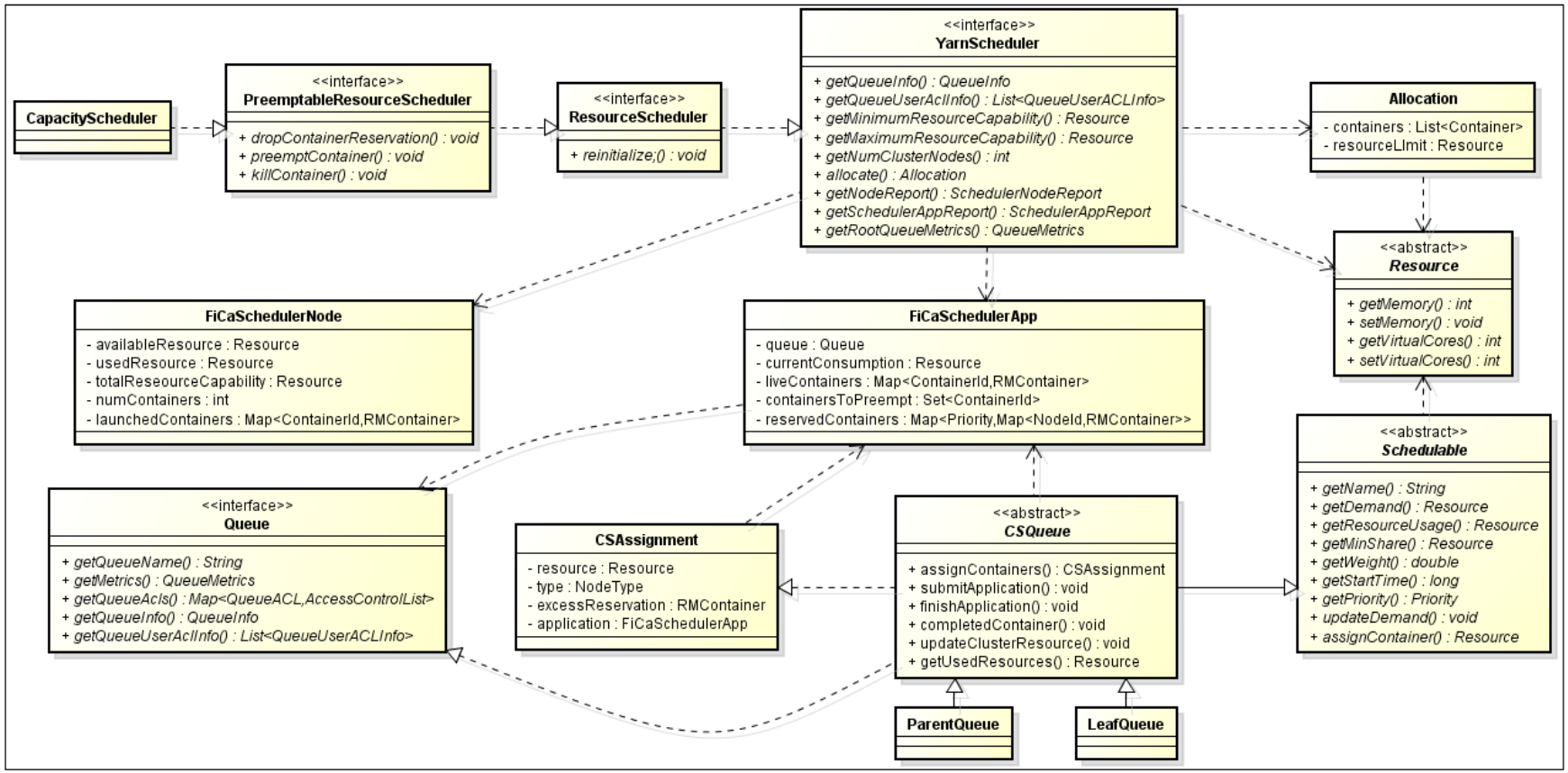
(HortonWorks,2014)

Estudo da Arquitetura



(HortonWorks,2014)

Estudo da Arquitetura



Estudo da política alocação

Valores do **cluster** para definir limites de alocação.

- XML:
  - *yarn.scheduler.minimum-allocation-mb* (1024);
  - *yarn.scheduler.minimum-allocation-vcores* (1);
  - *yarn.scheduler.maximum-allocation-mb* (8192);
  - *yarn.scheduler.maximum-allocation-vcores* (32).
- Código:
  - **minimumAllocation**;
  - **maximumAllocation**.

informáticaufsm
Ciência da Computação

Estudo da política alocação

Valores passados pela **aplicação**.

- XML:
  - *mapreduce.map.memory.mb* (1024);
  - *mapreduce.map.cpu.vcores* (1);
  - *mapreduce.reduce.memory.mb* (1024);
  - *mapreduce.reduce.cpu.vcores* (1).
- Código:
  - **JobConf**.

informáticaufsm
Ciência da Computação

Experimento

Realizado para melhor entendimento da política de alocação.
Foram utilizados 4 cenários de requisição:

- *default*, nenhum valor alterado;
- maior que máximo, o limite máximo foi alterado;
- menor que mínimo, limite mínimo foi alterado;
- dentro dos limites.

informáticaufsm
Ciência da Computação

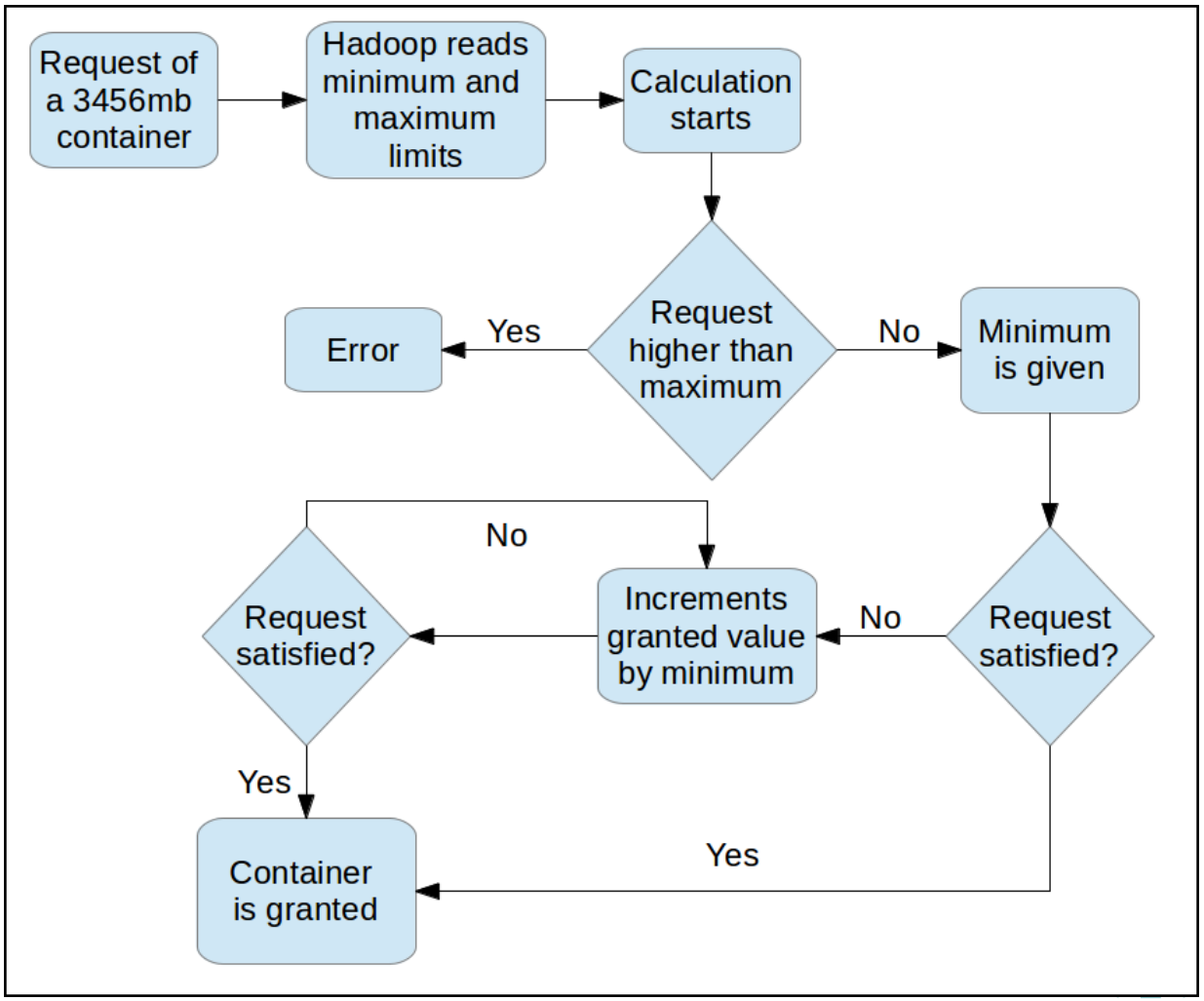
Experimento

Resultados:

Parâmetros	*Default*	Maior	Menor	Dentro
Requisição *Map* (mb)	1024	1024	1024	3456
Requisição *Reduce* (mb)	1024	1024	1024	3712
Limite mínimo (mb)	1024	512	2048	512
Limite máximo (mb)	8192	768	8192	8192
Alocação *Map* (mb)	1024	ERRO	2048	3584
Alocação *Reduce* (mb)	1024	ERRO	2048	4096

informáticaufsm
Ciência da Computação

Experimento - cálculo



informáticaufsm
Ciência da Computação

Escalonamento sensível ao contexto

- *CapacityScheduler* já provem uma infra-estrutura.
- Falhas/pontos fracos do funcionamento.
- Oportunidade de melhora - **Coletor**.

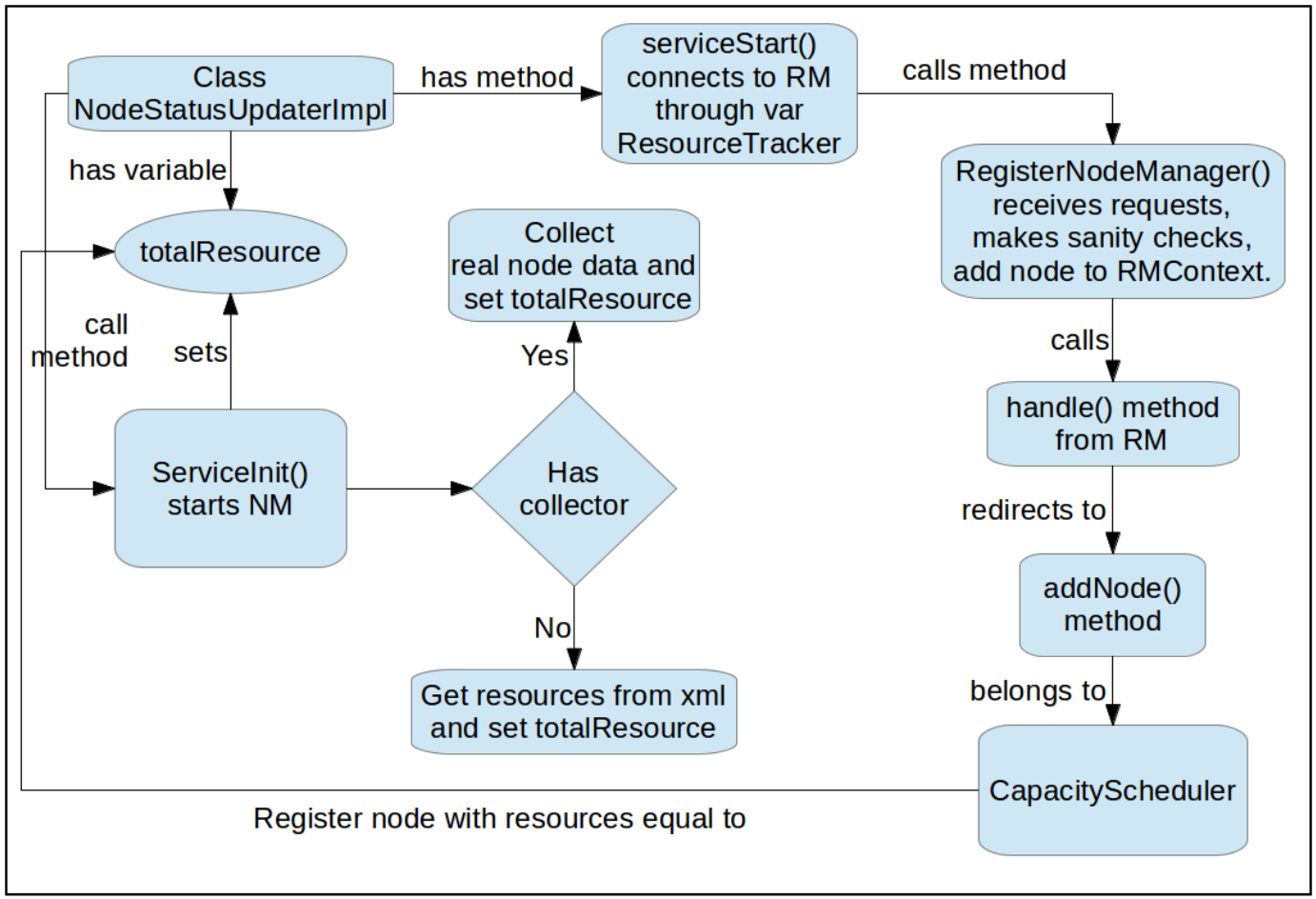
informáticaufsm
Ciência da Computação

Coletor

- Coletor escolhido: PER-MARE *Collector* (Kirsch-Pinheiro, M., 2013).
- Organizado em: interface, classe abstrata e classes de implementação.
- Integração com o Hadoop: estudo de como o *CapacityScheduler* obtinha informação sobre os recursos de cada *NodeManager*.

informáticaufsm
Ciência da Computação

Integração do Coletor com o Hadoop



Roteiro

Teste de integração do coletor

Realizado no Grid'5000, para testar se o coletor fazia a coleta corretamente.

- Conf. *Hardware*: 2 CPU AMD@1.7Ghz, 12 cores/CPU e 47GB RAM.
- Conf. *Software*: Ubuntu x64 12.04, Hadoop 2.2.0, Sun JDK 1.7.
- Conf. Hadoop: *default*.
  - *yarn.nodemanager.resource.memory-mb* (8192)
  - *yarn.nodemanager.resource.cpu-vcores* (8)

informáticaufsm
Ciência da Computação

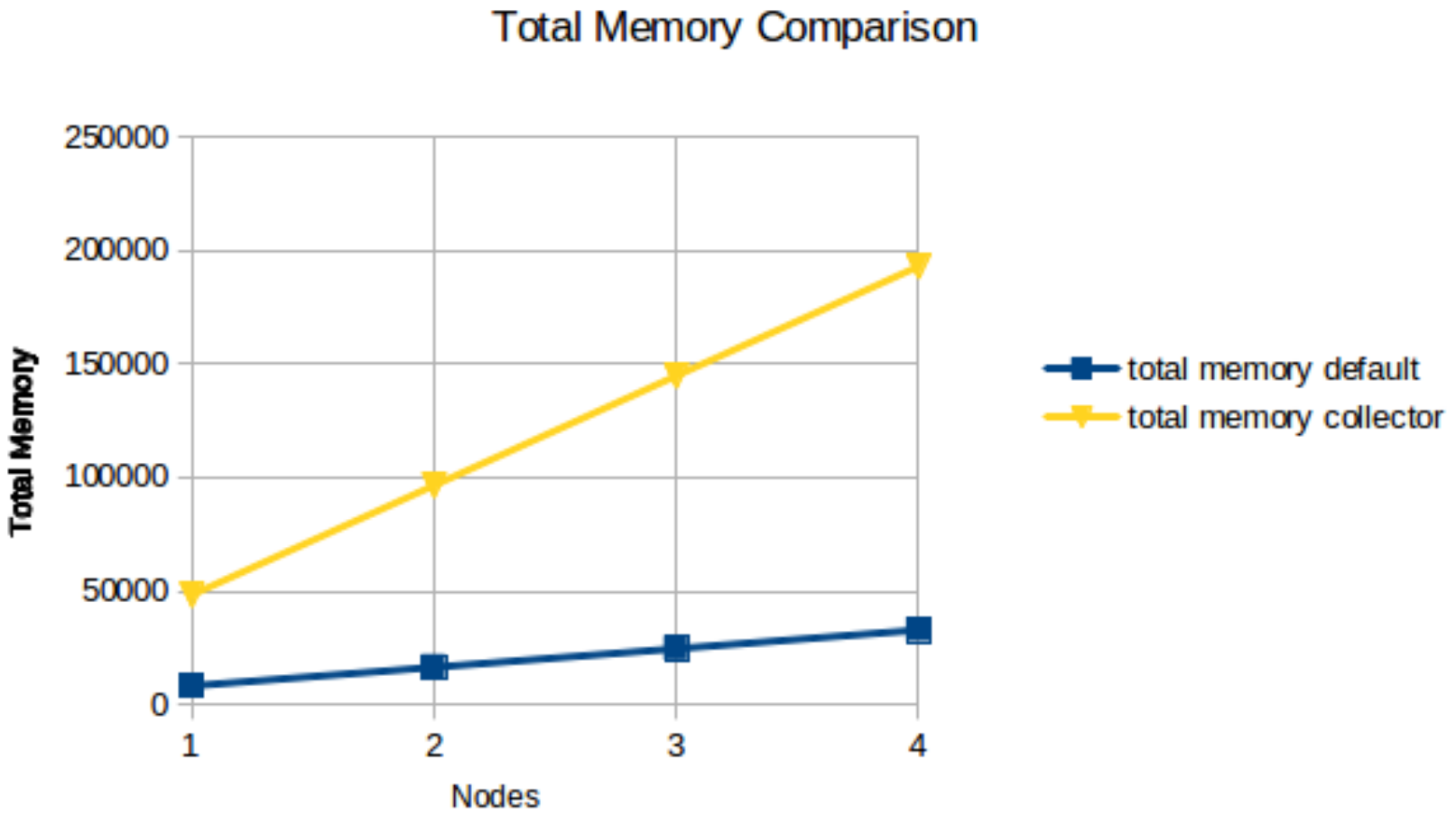
Teste de integração do coletor

Recursos disponíveis por *NodeManager* utilizando configuração *default* e utilizando coletores de informação:

	Default	Coletor
Node Memory	8192	48303
Node Vcores	8	24

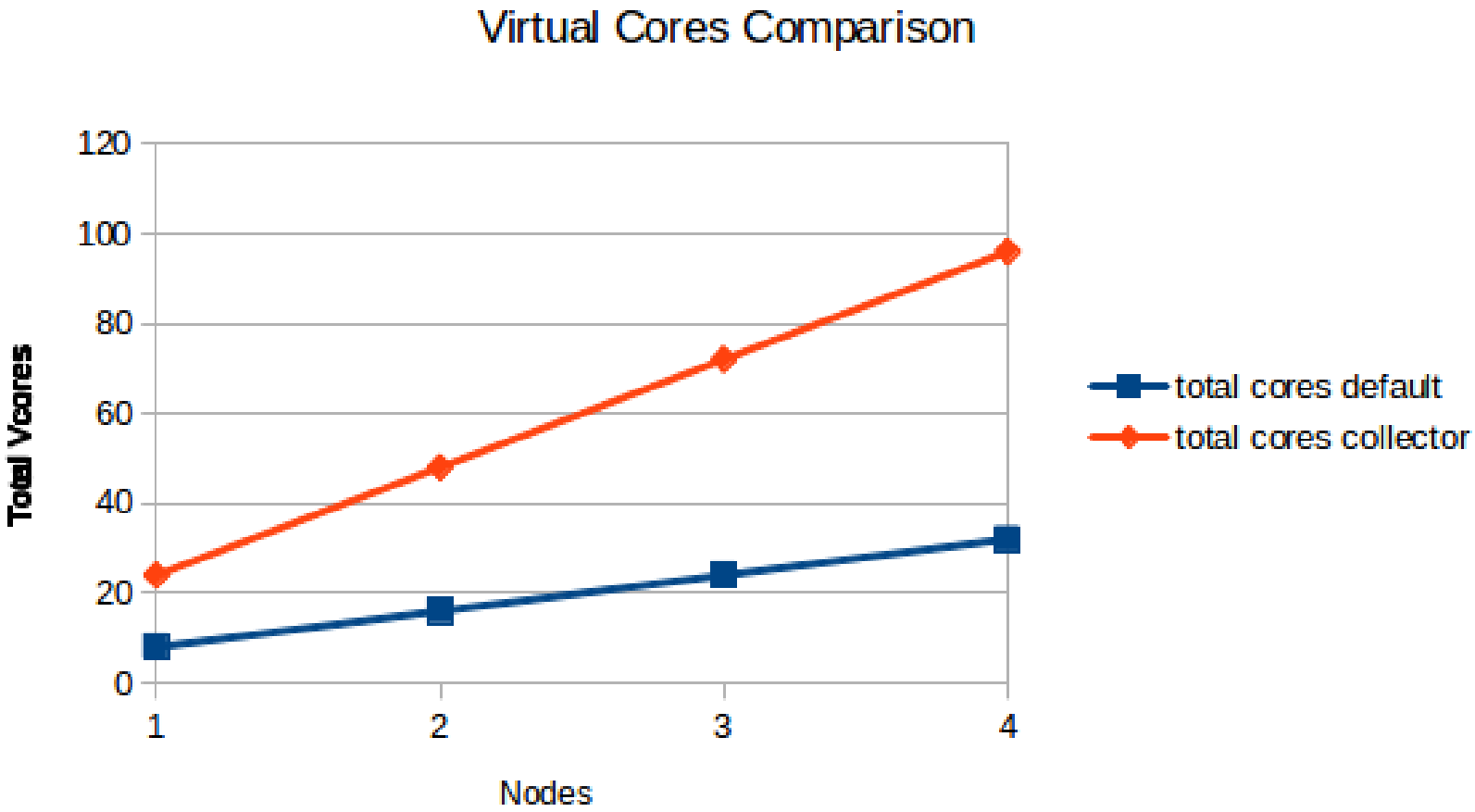
Teste de integração do coletor

Recursos disponíveis no cluster utilizando configuração *default* e utilizando coletores de informação:



Teste de integração do coletor

Recursos disponíveis no cluster utilizando configuração *default* e utilizando coletores de informação:



Teste de execução: CapacityScheduler original X CapacityScheduler sensível ao contexto

Realizado no Grid'5000, para comparar o funcionamento do *CapacityScheduler* original com o *CapacityScheduler* sensível ao contexto.

- Configurações de *Hardware*, *Software* e Hadoop iguais ao experimento anterior.
- Aplicação utilizada para *benchmark*: *TeraSort* de 5GB.

informáticaufsm
Ciência da Computação

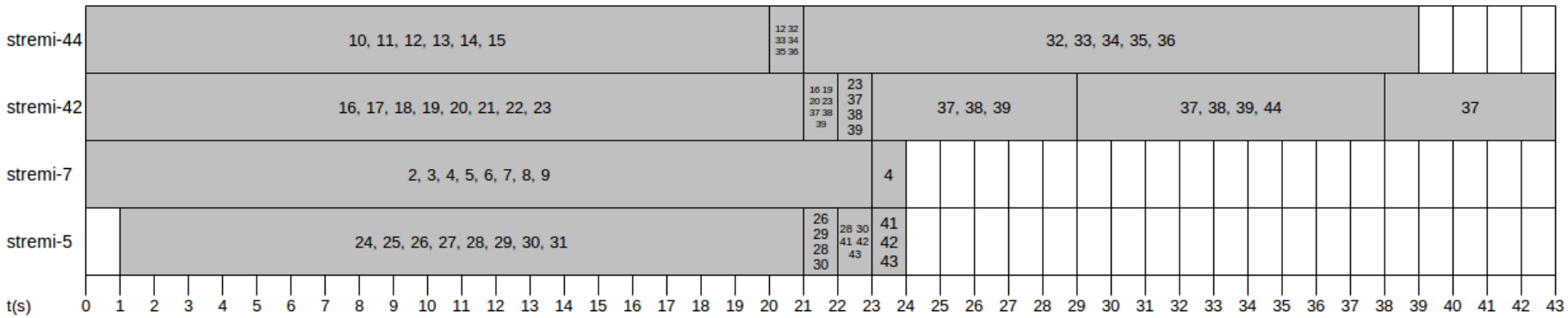
Teste de execução: CapacityScheduler original X CapacityScheduler sensível ao contexto

Resultados referentes aos limites de alocação e disponibilidade de recursos:

- *CapacityScheduler* original:
  - recursos totais: 32768 mb e 32 cores;
  - limite mínimo: 1024 mb e 1 core;
  - limite máximo: 8192 mb e 8 cores.
  - todos *containers* de *map* receberam 1024 mb e 1 core, o limite mínimo.
- *CapacityScheduler* sensível ao contexto:
  - recursos totais: 193210 mb e 96 cores;
  - limite mínimo: 4830 mb e 2 cores;
  - limite máximo: 24151 mb e 12 cores.
  - todos *containers* de *map* receberam 4830 mb e 2 cores, o limite mínimo.

Teste de execução: CapacityScheduler original X CapacityScheduler sensível ao contexto

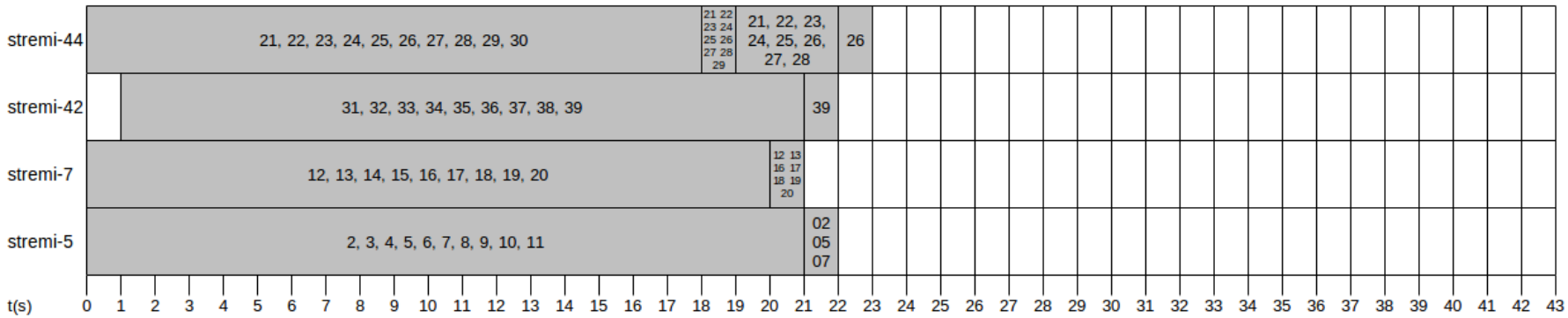
Resultados referentes ao escalonamento dos recursos: apresentados em forma de gráfico de Gantt.



informáticaufsm
Ciência da Computação

Teste de execução: CapacityScheduler original X CapacityScheduler sensível ao contexto

Resultados referentes ao escalonamento dos recursos: apresentados em forma de gráfico de Gantt.



informáticaufsm
Ciência da Computação

Roteiro

informáticaufsm
Ciência da Computação

Conclusão

- O *CapacityScheduler* é capaz de detectar e reagir conforme a informação do ambiente (memória e cpu dos *NodeManagers*).
- O escalonamento foi melhorado.
- Existem mais formas de melhorar o escalonamento.

informáticaufsm
Ciência da Computação

Trabalhos Futuros

- Extender a classe *Resources* para suportar outras informações de recurso como carga de cpu, espaço em disco disponível, entre outras.
- Melhorar o escalonamento do *CapacityScheduler* / criar um novo escalonador que possa levar em consideração outros recursos.
- Modificar o *ApplicationMaster*.

informáticaufsm
Ciência da Computação

Roteiro

informáticaufsm
Ciência da Computação

Referências

- DEY, A. K. Understanding and Using Context. Personal Ubiquitous Comput., London, UK, UK, v.5, n.1, p.4-7, Jan. 2001.
- MAAMAR, Z.; BENSLIMANE, D.; NARENDRA, N. C. What can context do for web services? Commun. ACM, New York, NY, USA, v.49, n.12, p.98-103, Dec. 2006.
- KUMAR, K. A. et al. CASH: context aware scheduler for hadoop. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS, New York, NY, USA. Proceedings. . . ACM, 2012. p.52-61. (ICACCI '12).
- ZAHARIA, M. et al. Improving MapReduce performance in heterogeneous environments. In: USENIX CONFERENCE ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION, 8., Berkeley, CA, USA. Proceedings. . . USENIX Association, 2008. p.29-42. (OSDI'08).
- TIAN, C. et al. A Dynamic MapReduce Scheduler for Heterogeneous Workloads. In: EIGHTH INTERNATIONAL CONFERENCE ON GRID AND COOPERATIVE COMPUTING, 2009., Washington, DC, USA. Proceedings. . . IEEE Computer Society, 2009. p.218-224. (GCC '09).
- CHEN, Q. et al. SAMR: a self-adaptive mapreduce scheduling algorithm in heterogeneous environment. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, 2010., Washington, DC, USA. Proceedings. . . IEEE Computer Society, 2010. p.2736-2743. (CIT '10).

Referências (Continuação)

- RASOOLI, A.; DOWN, D. G. Coshh: a classification and optimization based scheduler for heterogeneous hadoop systems. In: SC COMPANION: HIGH PERFORMANCE COMPUTING, NETWORKING STORAGE AND ANALYSIS, 2012., Washington, DC, USA. Proceedings. . . IEEE Computer Society, 2012. p.1284-1291. (SCC '12).
- ISARD, M. et al. Quincy: fair scheduling for distributed computing clusters. In: ACM SIGOPS 22ND SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, New York, NY, USA. Proceedings. . . ACM, 2009. p.261-276. (SOSP '09).
- XIE, J. et al. Improving MapReduce performance through data placement in heterogeneous Hadoop clusters. In: PARALLEL AND DISTRIBUTED PROCESSING, WORKSHOPS AND PHD FORUM (IPDPSW). Anais. . . IEEE International Symposium, 2010.
- HADOOP, A. Arquitetura do HDFS. http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/Federation.html, Acesso em novembro de 2013.
- HADOOP, A. Arquitetura do YARN. http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html, Acesso em novembro de 2013.

Referências (Continuação)

- HortonWorks. HortonWorks Hadoop YARN - ResourceManager. http://hortonworks.com/blog/apache-hadoop-yarn-resourcemanager/, acesso em janeiro de 2014.
- HortonWorks. HortonWorks Hadoop YARN - NodeManager. http://hortonworks.com/blog/apache-hadoop-yarn-nodemanager/, acesso em janeiro de 2014.

Desenvolvimento de um Escalonador Sensível ao Contexto para o *Apache Hadoop*

Guilherme Weigert Cassales[1]
Orientadora: Profª Drª Andrea Schwertner Charão[1]

[1]Ciência da Computação
Universidade Federal de Santa Maria

20 de janeiro de 2014