Escalonamento de Processos Sensível à Localidade de Dados em Sistemas de Arquivos Distribuídos

Bruno Hott
Rodrigo Rocha
Dorgival Guedes

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

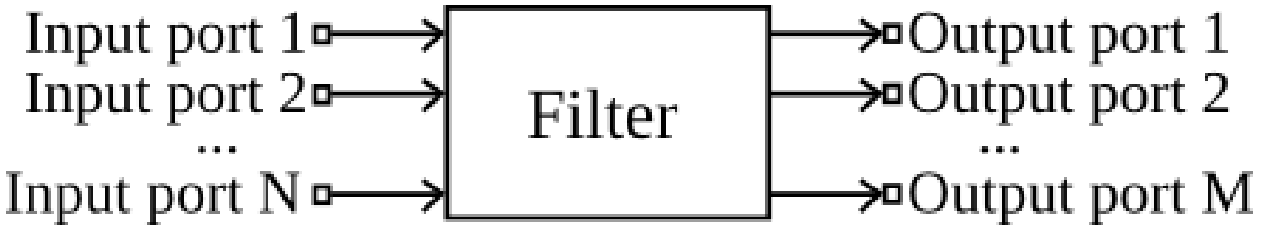
UFMG

Introdução

- Volume muito grande de dados distribuídos por diversas máquinas
- Mover as aplicações para perto das bases de dados de forma eficaz
- Hadoop HDFS considera localidade a nível de rack
- Impacto da localidade dos dados a nível de máquina nesses frameworks

Watershed

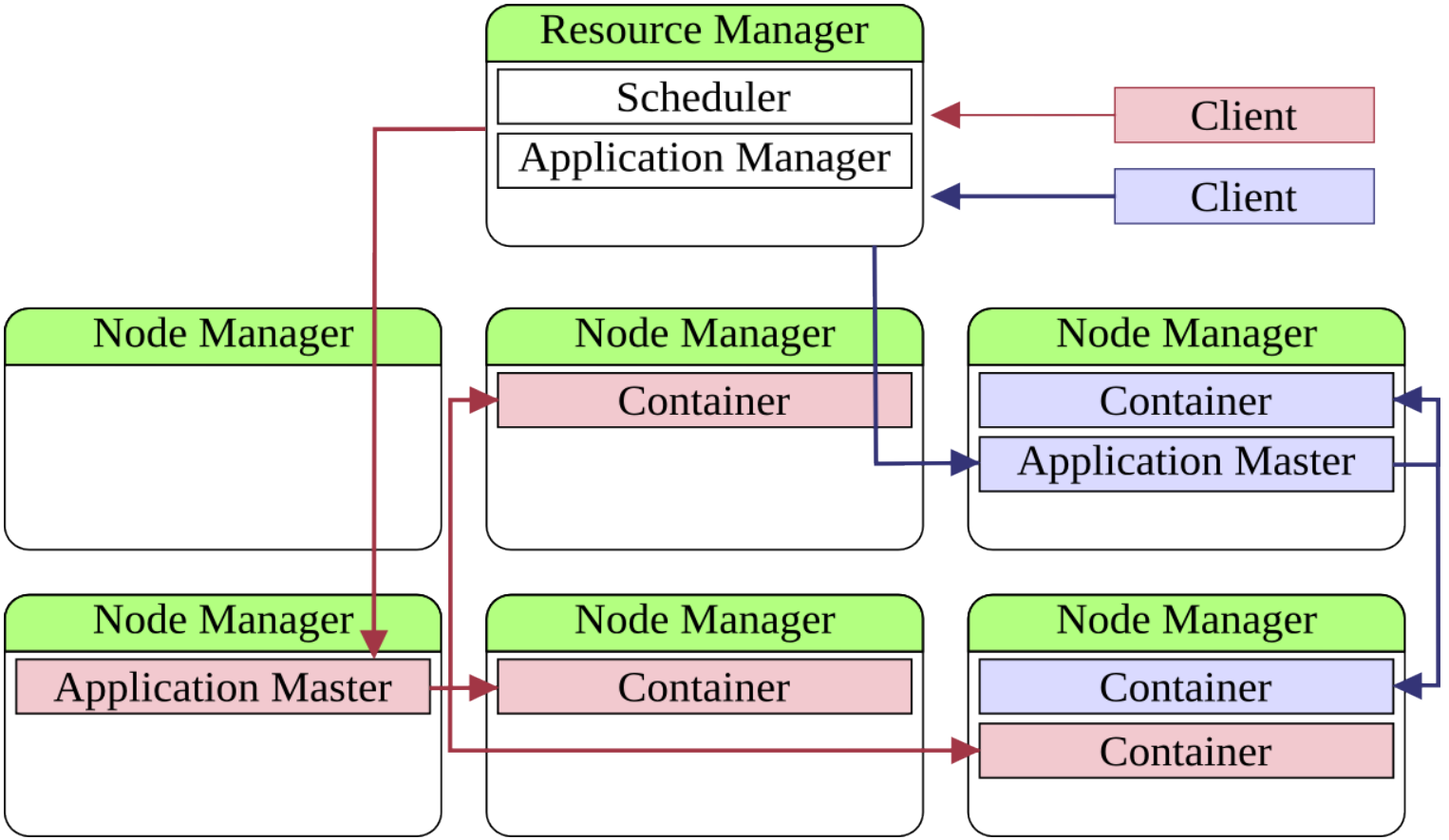
- Framework para processamento distribuído que implementa a abstração filtro-fluxo
- Dados fluem através de fluxos e são processados por filtros
- Cada filtro pode ter múltiplas instâncias de execução
- Integrado com: YARN, HDFS/Tachyon



Hadoop YARN

- Plataforma para gerenciamento de aplicações distribuídas
- Permite que diferentes aplicações possam coexistir em um mesmo ambiente
- Dividido em:
  - Gerenciamento de recursos (RM)
  - Escalonamento de tarefas (AM)
- Composto por:
  - *ResourceManager* (RM)
  - *NodeManager* (NM)

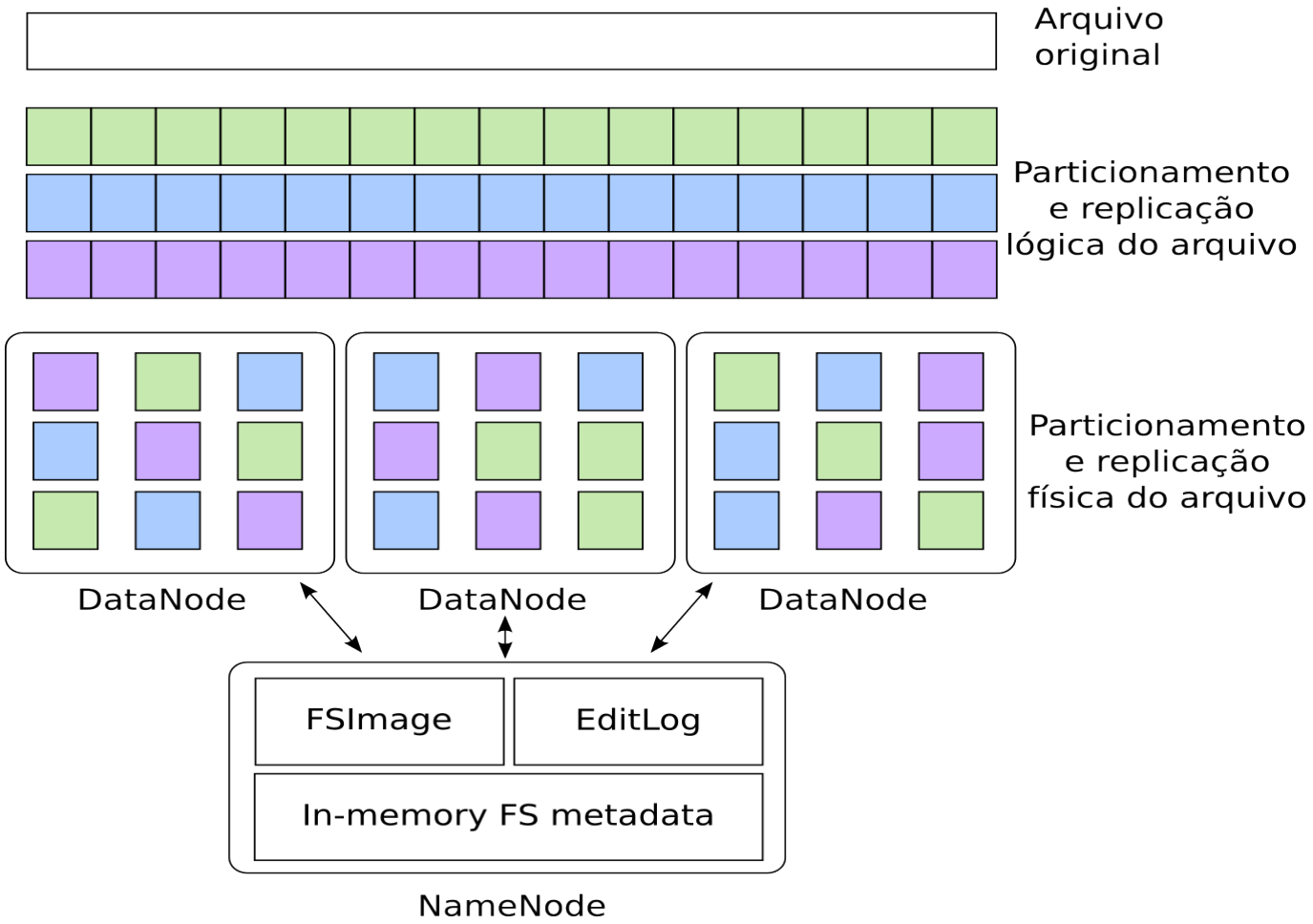
Hadoop YARN


Resource Manager
Scheduler
Application Manager
Client
Client
Node Manager
Node Manager
Container
Node Manager
Container
Application Master
Node Manager
Application Master
Node Manager
Container
Node Manager
Container
Container

Hadoop HDFS

- Sistema distribuído de arquivos
- Replicação garante:
  - Durabilidade
  - Largura de banda
- Composto por:
  - *NameNode* (NN)
  - *DataNodes* (DN)

Hadoop HDFS


Arquivo original
Particionamento e replicação lógica do arquivo
Particionamento e replicação física do arquivo
DataNode
DataNode
DataNode
FSImage
EditLog
In-memory FS metadata
NameNode

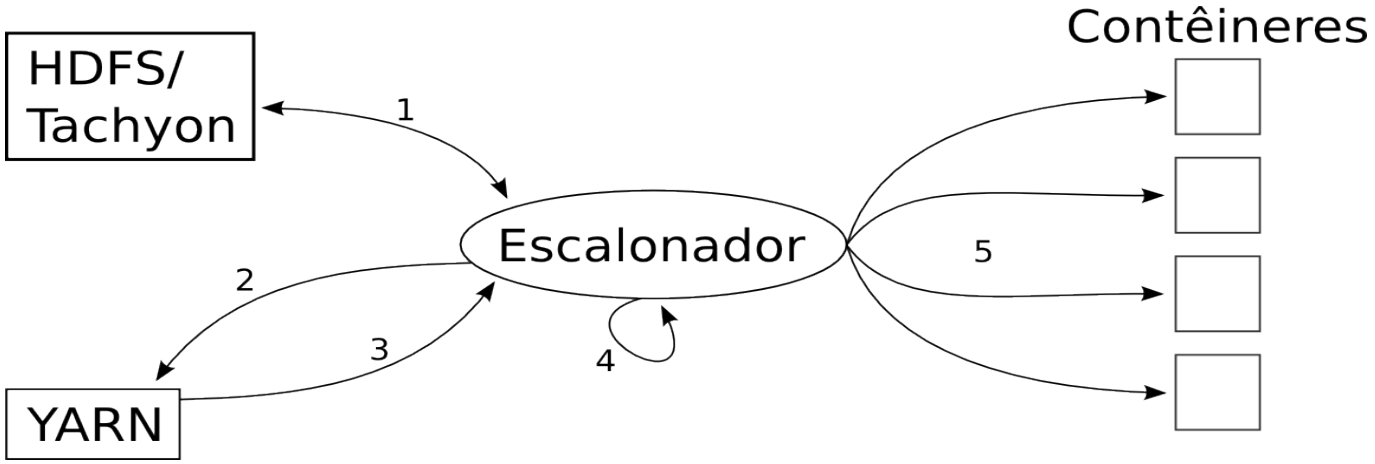
Tachyon (Alluxio)

- Cache para sistemas distribuídos de arquivos
- Interface espelha com HDFS
- Composto por:
  - *Master*
  - *Workers*

Escalonador Sensível à Localidade dos Dados

- Consulta ao sistema de arquivos (HDFS/Tachyon)
- Localização de cada bloco
- Escalonamento em duas fases:
  - Blocos locais
  - Blocos remotos
- Alocação dos contêineres (YARN)

Escalonador Sensível à Localidade dos Dados



1. Busca do arquivo no HDFS/Tachyon com lista de blocos
2. Pedido de contêineres
3. Recebimento dos contêineres fora de ordem
4. Organização dos contêineres em função dos blocos
5. Execução dos processos da aplicação

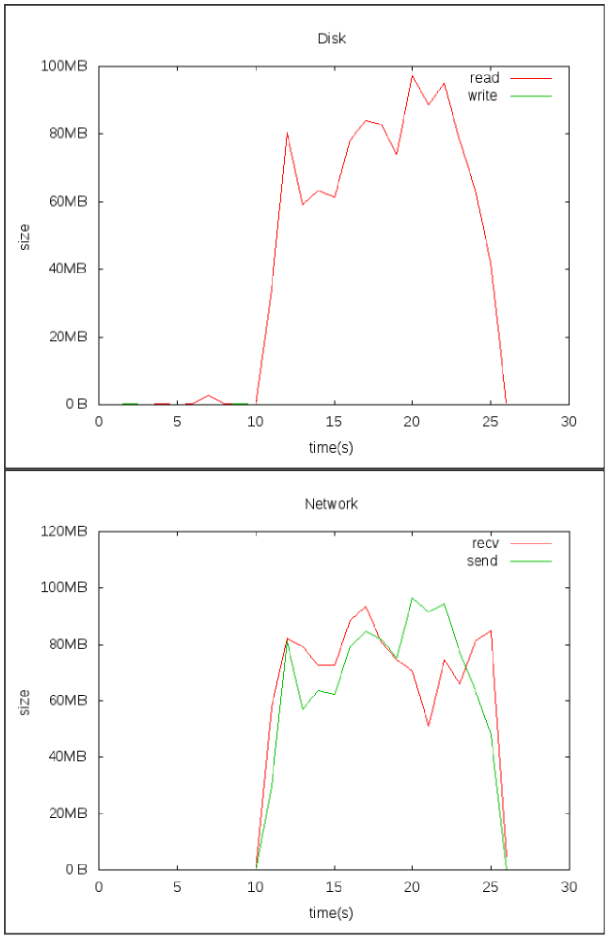
Avaliação Experimental

- Micro-benchmark: leitura simples
- Políticas implementadas
  - `sec:` Escalonamento YARN
  - `esc:` Escalonamento proposto
  - `ale:` Escalonamento aleatório
  - `pes:` Pior caso
- 10 VMs: 2 VCPUs (2,5GHz) e 4GB RAM
- 5 execuções

Avaliação Experimental

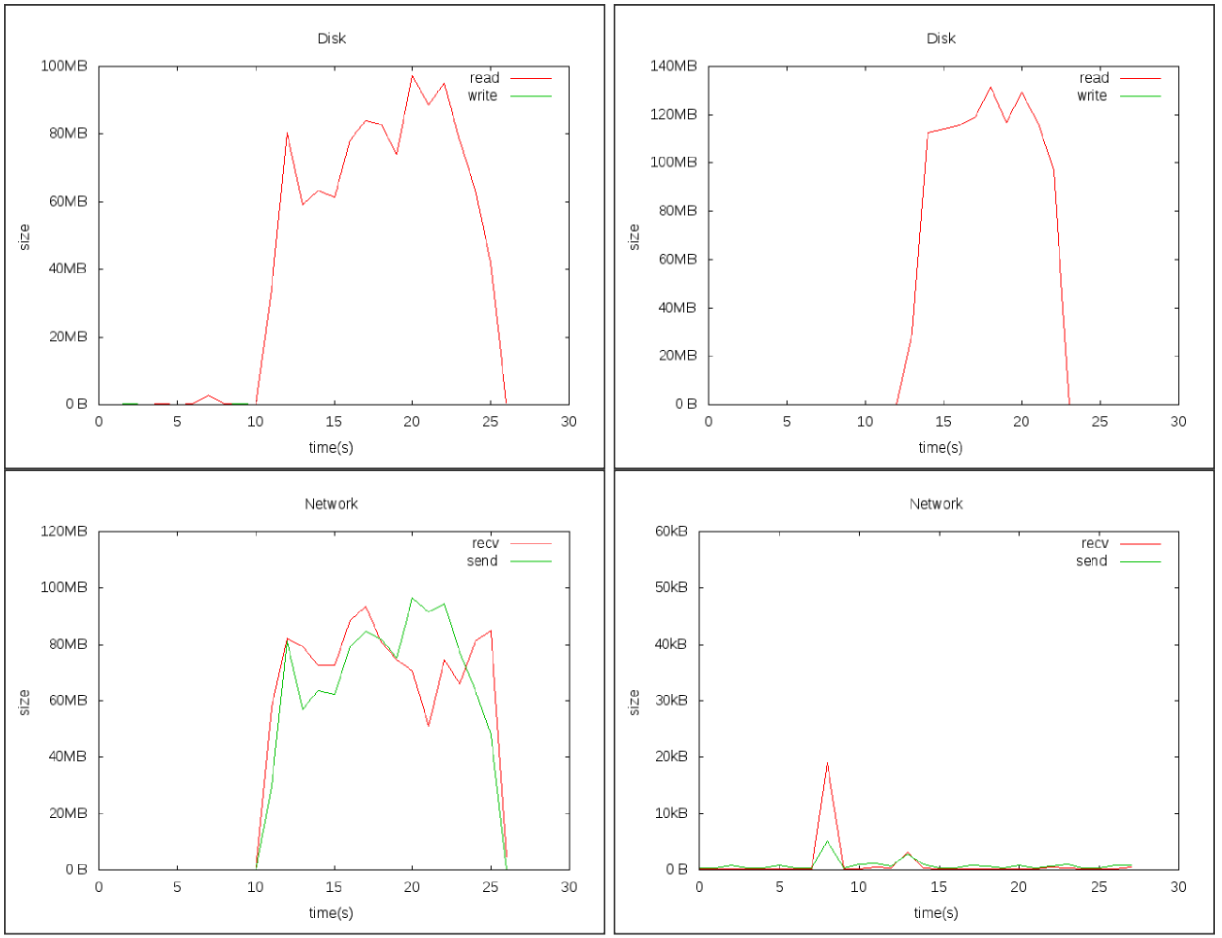
- Fator de replicação 2
- Limpeza de cache:
  - HDFS (disco)
  - Tachyon (memória)
- Grandezas de interesse:
  - Tempo de execução
  - Tamanho do cluster
  - Porção de dados por máquina

Avaliação Experimental: Validação



(a) Execução sem localidade

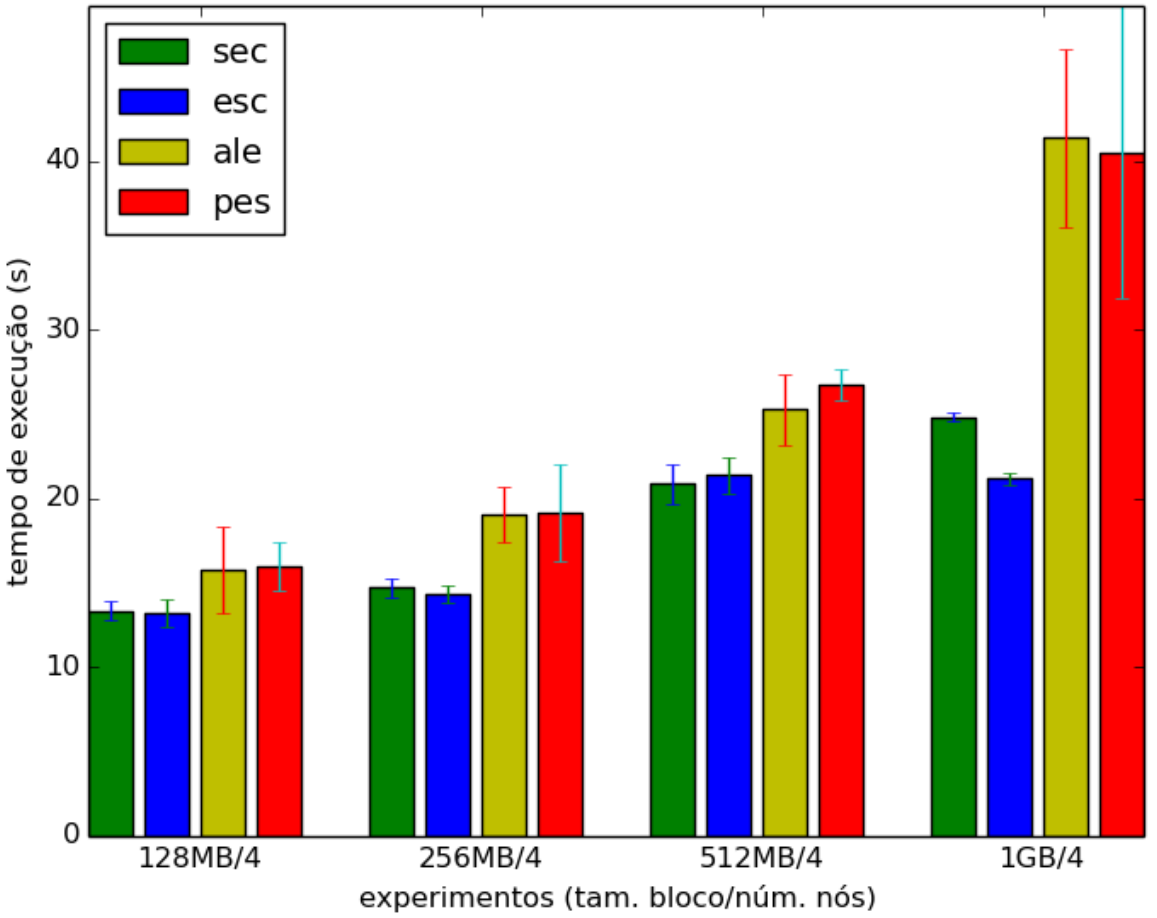
Avaliação Experimental: Validação



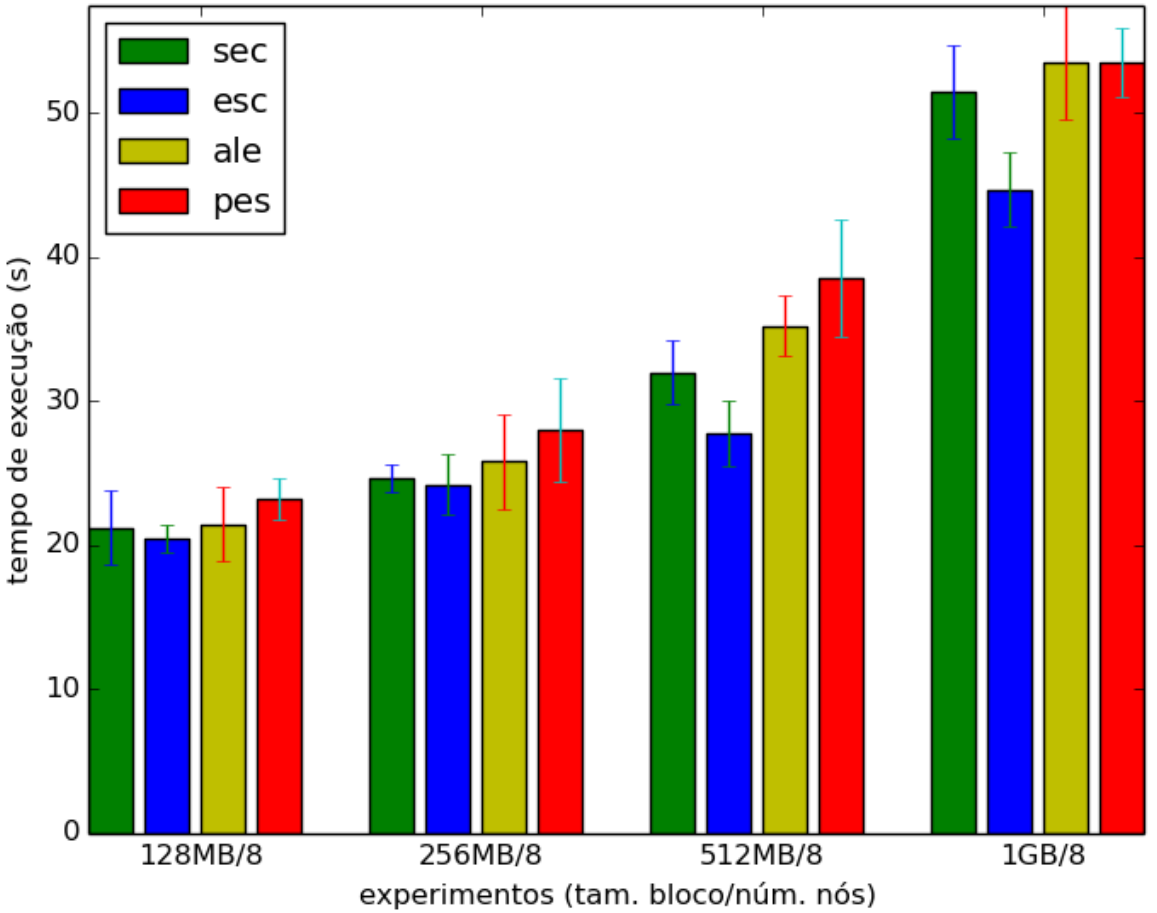
(a) Execução sem localidade

(b) Execução com o escalonador proposto

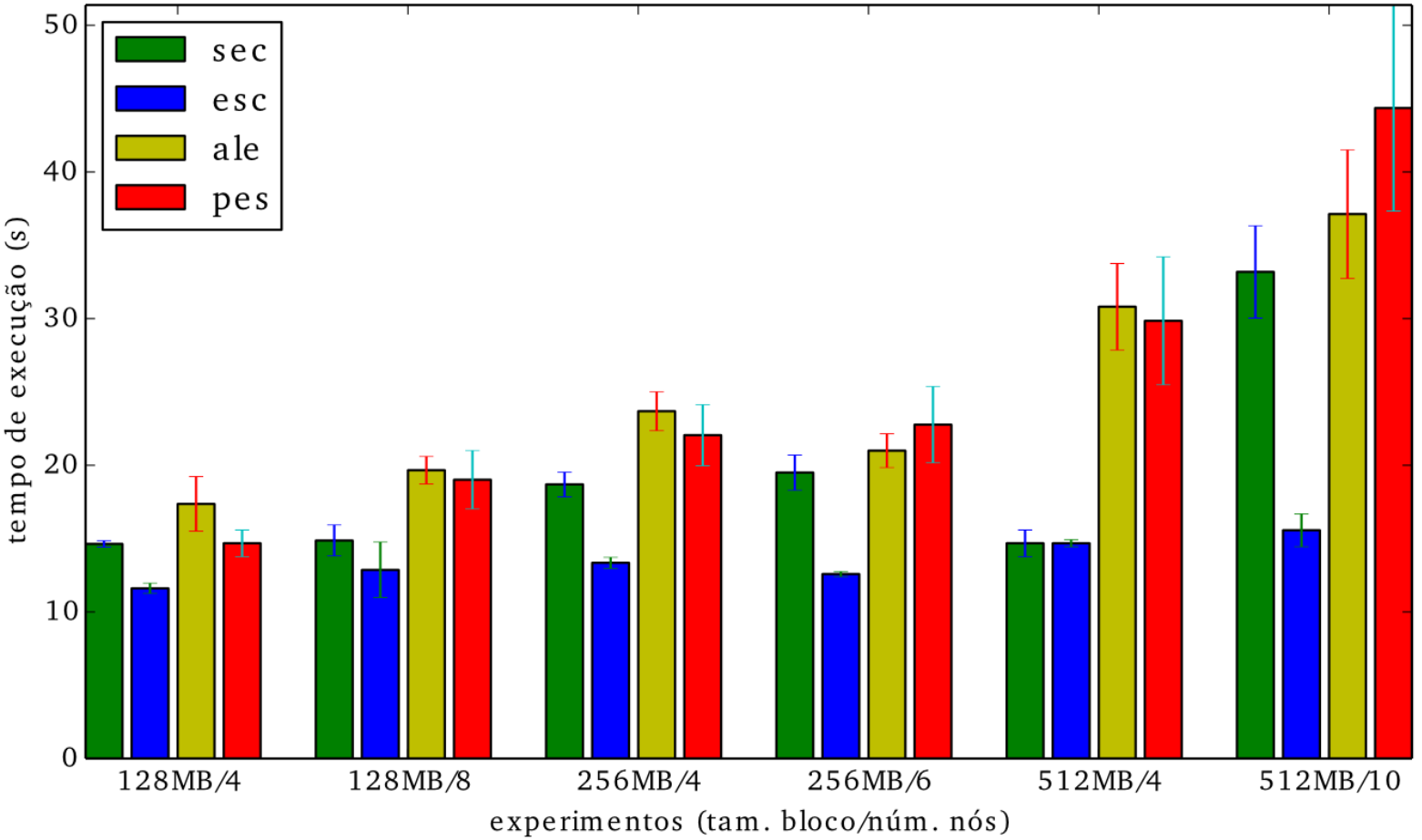
Avaliação Experimental: HDFS (4 nós)



Avaliação Experimental: HDFS (8 nós)



Avaliação Experimental: Tachyon



Conclusão

- HDFS: localidade a **nível de rack**
- Avaliamos localidade a **nível de máquina**
- Desenvolvemos um escalonador sensível à localidade dos dados integrado ao HDFS e ao YARN
- Escalonamento sensível à localidade beneficia significativamente a execução
- Ganhos ainda mais significativos quando os dados já estão em memória (Tachyon)

Trabalhos Futuros

- O escalonador proposto é facilmente extensível dentro do ecossistema Hadoop
- Portar para Hadoop MapReduce e Spark
- Estender a política de escalonamento para incluir heurísticas para casos de alocação em cenários multiusuários com maior contenção de recursos

Obrigado!

PERGUNTAS?

brhott@dcc.ufmg.br