Avaliação de sistemas NoSQL para o gerenciamento de dados em *workflows* científicos

Gabriel Ferreira Guilhoto Orientadora: Profa. Dra. Kelly Rosa Braghetto

Trabalho de Formatura Supervisionado (MAC0499) Novembro de 2015
Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP)

Introdução

Workflows científicos são automações de experimentos científicos que, por natureza, geralmente manipulam grandes quantidades de dados. Para lidarem com os dados consumidos e produzidos, geralmente recorrem a sistemas de arquivos distribuídos, a sistemas de armazenamento baseados em objetos ou à eliminação de resultados intermediários, que nem sempre satisfazem a todos os requisitos dos cientistas. Sistemas NoSQL podem ser mais adequados para esse fim, por disporem de mecanismos como a replicação e o particionamento de dados para melhorar a escalabilidade e a disponibilidade e por poderem servir como um repositório de dados compartilhado.

Este trabalho avaliou o uso de sistemas NoSQL no gerenciamento de dados em workflows científicos. Para isso, foram consideradas operações de manipulações de dados frequentemente encontradas em aplicações científicas, tais como filtragem, distribuição, agregação e processamento de janelas de dados.

Sistemas NoSQL

Os sistemas NoSQL surgiram como alternativa aos sistemas gerenciadores de bancos de dados relacionais, oferecendo maior capacidade para lidar com grandes volumes de dados, melhor desempenho em operações simples de manipulação dos dados e modelos de representação de dados mais flexíveis. Geralmente são projetados para serem executados em ambientes distribuídos, e para isso adotam estratégias para a replicação e o particionamento dos dados em várias máquinas, o que possibilita paralelismo no acesso aos dados.

Neste trabalho, foram avaliados dois sistemas NoSQL diferentes: o MongoDB e o Cassandra.

- **MongoDB**:
  - Orientado a documentos; armazena seus dados em formato JSON sem estrutura pré-definida.
  - Utiliza a configuração mestre-escravo para a replicação; todas as operações de escrita são direcionadas a um nó mestre e depois replicadas nos nós escravos.
  - Pode-se permitir ler dos escravos para melhor desempenho.
  - Para reduzir o número de operações em cada máquina, é feito o particionamento dos dados em vários servidores.
  - Cada partição é um conjunto de réplicas que é responsável por um subconjunto dos dados.
- **Cassandra**:
  - Orientado a famílias de colunas; armazena dados como linhas com muitas colunas associadas.
  - A replicação é do tipo ponto a ponto; não há um nó mestre e todos os nós podem atender a leituras e escritas.
  - Cada linha inserida é primeiramente armazenada em um nó determinado pela chave e depois replicada nos próximos nós em sentido horário, em quantidade definida por um fator de replicação.

Workflows Científicos

Um *workflow* científico é uma descrição de um processo computacional de análise de dados. *Workflows* científicos são tipicamente modelados como fluxos de dados, ou seja, são compostos por atividades que produzem dados que podem ser consumidos por outras atividades. São representados por um digrafo acíclico em que um vértice é uma atividade e um arco $u \to v$ determina que a atividade u produz dados a serem consumidos por v. Existem sistemas gerenciadores de *workflows* científicos (SGWCs) capazes de apoiar o ciclo de vida de um *workflow* científico, permitindo defini-los, executá-los e monitorá-los. No trabalho, foi usado o SGWC Pegasus, que permite a execução de um *workflow* em recursos heterogêneos distribuídos.

Experimentos

Foram feitos experimentos com o MongoDB e o Cassandra com base num *workflow* científico criado por Elaine Naomi Watanabe para o Pegasus. Esse *workflow* faz análise de dados de rastros de um *cluster* do Google com cerca de 500 mil registros.

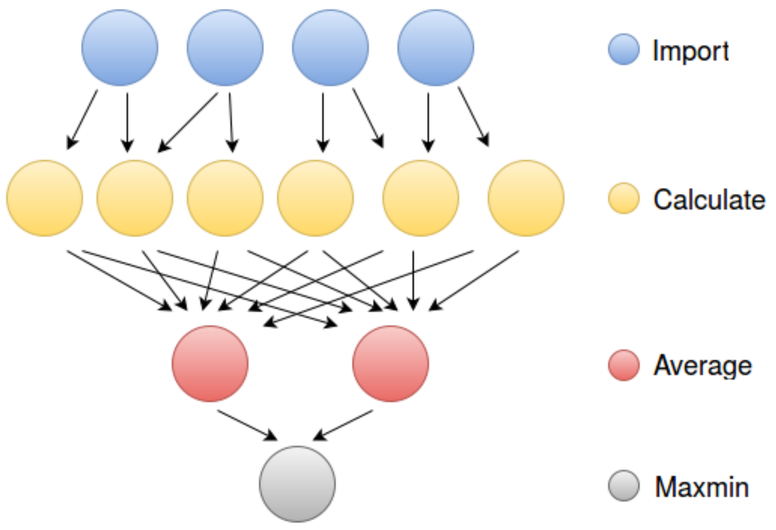

Figura 1: Representação simplificada do *workflow* usado nos experimentos. No *workflow* real as atividades são em maior número.

O *workflow* recebe arquivos CSV com dados sobre execuções de tarefas (e.g., consumo de CPU e memória) no *cluster* e tem três tipos de atividade:

- `Import`: insere os dados do CSV no NoSQL
- `Calculate`: calcula a razão entre os consumos de CPU e memória das tarefas de um subconjunto dos dados
- `Average`: calcula as médias das razões entre CPU e memória para as tarefas de um determinado tipo
- `Maxmin`: determina os tipos de tarefa com maior e menor razão entre consumo de CPU e memória

Todas as atividades, com exceção das do tipo `Import`, leem dados de entrada e gravam dados de saída no sistema NoSQL.

Foram usadas dez máquinas da Nuvem USP, cada uma com 32 GB de memória RAM, 8 núcleos a 2300 MHz e SO Ubuntu Server 12.04 ou Ubuntu 14.04.

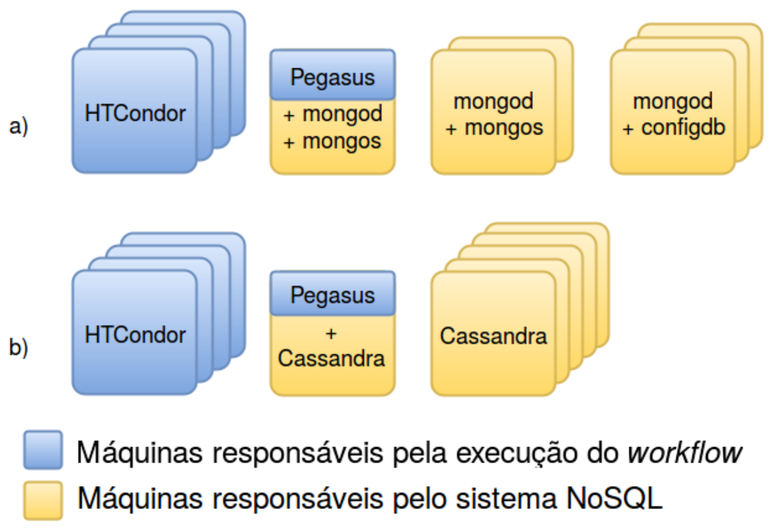

Figura 2: Responsabilidades de cada máquina no experimento com a) o MongoDB e b) o Cassandra

As máquinas com o HTCondor são as responsáveis por executar o *workflow* propriamente dito, e o Pegasus monitora o estado das atividades para determinar qual pode executar em cada momento. Como não faz processamento pesado, a máquina em que o Pegasus está é compartilhada com o sistema NoSQL.

No MongoDB, os `mongod` armazenam os dados da aplicação, os `configdb` têm metadados sobre a partição em que cada documento se encontra e os `mongos` recebem as conexões de clientes. No Cassandra, as máquinas têm todas a mesma responsabilidade.

Resultados e Conclusão

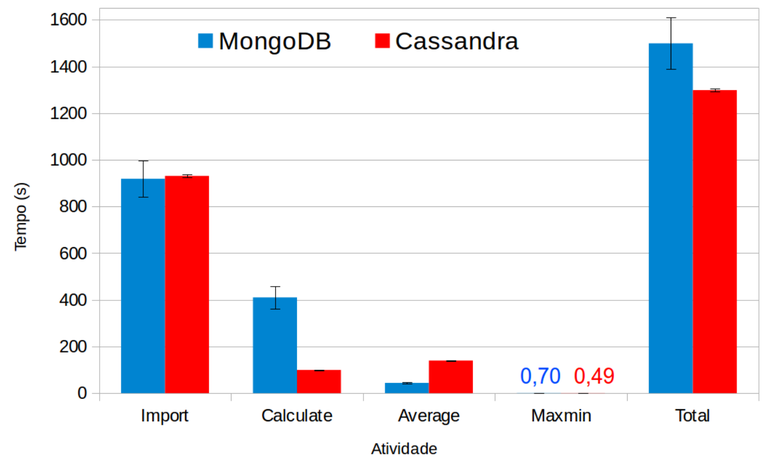

Figura 3: Tempos das atividades de acordo com o sistema NoSQL

O *workflow* foi executado trinta vezes para cada sistema NoSQL, e foram medidos, para cada tipo de atividade, o tempo desde o início da primeira até o término da última. No gráfico da figura 3 estão representadas as médias de tempo de execução com o intervalo de confiança de 95%. A tabela abaixo indica qual sistema teve desempenho melhor de acordo com o tipo de operação das atividades. As atividades `Maxmin` não foram consideradas pois o tempo de execução é insuficiente para se tirar conclusões.

Tipo de atividade	**Tipo de operação**	**Sistema com melhor desempenho**
`Import`	Inserção	Empate
`Calculate`	Seleção por faixa de valores e inserção	Cassandra
`Average`	Seleção por igualdade e inserção	MongoDB

Percebe-se que os dois sistemas têm desempenho semelhante nas operações de inserção (escrita) de dados, embora o Cassandra leve vantagem na seleção por faixa de valores e o MongoDB seja mais rápido na seleção por igualdade em um valor.

Referências Principais

[Apa] The Apache Software Foundation.
The Apache Cassandra Project.
http://cassandra.apache.org/.

[BC14] Kelly Rosa Braghetto e Daniel Cordeiro.
Introdução à modelagem e execução de workflows científicos.
Atualizações em Informática. 1ª ed. Porto Alegre: SBC, páginas 1–40, 2014.

[DSS+05] Ewa Deelman, Gurmeet Singh, Mei-Hui Su, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, G Bruce Berriman, John Good, et al.
Pegasus: A framework for mapping complex scientific workflows onto distributed systems.
Scientific Programming, 13(3):219–237, 2005.

[FS] Martin Fowler e Pramod J. Sadalage.
NoSQL essencial: um guia conciso para o mundo emergente da persistência poliglota.
Novatec, 1ª edição.

[Mon] MongoDB, Inc.
MongoDB.
https://www.mongodb.org/.

[RWH11] Charles Reiss, John Wilkes, e Joseph L. Hellerstein.
Google cluster-usage traces: format + schema.
Technical report, Google Inc., Mountain View, CA, USA, novembro 2011.
http://code.google.com/p/googleclusterdata/wiki/TraceVersion2.