Centro Federal de Educação Tecnológica de Minas Gerais
Departamento de Computação
Engenharia de Computação

Comparação de algoritmos paralelos de ordenação em MapReduce

Mariane Raquel Silva Gonçalves
Orientadora: Cristina Duarte Murta

Belo Horizonte
17 de março de 2012

Sumário

1 INTRODUÇÃO

1.1 Tema do projeto

[Alguma citação que diga algo assim: com o surgimento das arquiteturas multicore, a computação paralela é hoje o caminho para desenvolver sistemas. Possivelmente Tally 2007, Sutter e Larus 2005 => sugiro os artigos [Asanovic et al. 2009]ASANOVIC, K. et al. A view of the parallel computing landscape. Commun. ACM e [Kale e Solomonik 2010] e mais alguns, veja em anexo] .

Uso crescente de computação paralela para sistemas computacionais gera a necessidade de algoritmos de ordenação inovadores, desenvolvidos para dar suporte a essas aplicações. A ordenação paralela é o processo de reorganizar uma sequência de entrada e produzir uma saída ordenada de acordo com um atributo através de múltiplas unidades de processamento, que ordenam em conjunto a sequência de entrada.

Na criação de algoritmos de ordenação paralela, é ponto fundamental ordenar coletivamente os dados de cada processo individual, de forma a utilizar todas as unidades de processamento e minimizar os custos de redistribuição de chaves entre os processadores (KALE; SOLOMONIK, 2010).

1.2 Relevância

O limite físico de aumento na frequência de operação dos processadores levou a indústria de hardware a substituir o processador de núcleo único por vários processadores eficientes em um mesmo *chip*, os processadores *multicore*. É preciso, então, criar aplicações que utilizem efetivamente o crescente número de núcleos dos processadores (ASANOVIC et al., 2009).

Um grande número de aplicações paralelas possui uma fase de computação intensa, na qual um lista de elementos deve ser ordenada com base em algum de seus atributos. Um exemplo é o algoritmo de Page Rank (PAGE et al., 1999) da Google: as páginas de resultado de uma consulta são rankeadas de acordo com sua relevância, e então precisam ser ordenadas de maneira eficiente (KALE; SOLOMONIK, 2010).

Fatores como movimentação de dados, balanço de carga, latência de comunicação e distribuição inicial das chaves são considerados ingredientes chave para o bom desempenho da ordenação paralela, e variam de acordo com o algoritmo escolhido como solução(KALE; SOLOMONIK, 2010). No exemplo do Page Rank, o número de páginas é enome, e elas são recolhidas de diversos servidores da Google; é uma questão fundamental escolher algoritmo paralelo com o melhor desempenho dentre as soluções possíveis.

Devido ao grande número de algoritmos de ordenação paralela e variadas arquiteturas paralelas, estudos experimentais assumem uma importância crescente na avaliação e seleçãoo de algoritmos apropriados para multiprocessadores.

O MapReduce(DEAN; GHEMAWAT, 2008) é um modelo de programação paralela criado pela Google para processamento e geração de grandes volumes de dados em *clusters*. Esse modelo propõe simplificar a computação paralela e ser de fácil uso, abstraindo conceitos complexos da paralelização - como tolerância a falhas, distribuição de dados e balanço de carga - e utilizando duas funções principais: map e reduce. A complexidade do algoritmo paralelo não é vista pelo desenvolvedor, que pode se ocupar em desenvolver a solução proposta. O Hadoop (WHITE, 2009) é uma das implementações do MapReduce, um *framework* open source que provê o gerenciamento de computação distribuída. Foi desenvolvido por Doug Cutting em 2005, e é amplamente apoiada e utilizado pela Yahoo!.

1.3 Objetivos

comparar duas ou mais implementações de algoritmos paralelos de ordenação pode citar o trabalho da Paula e dizer que é uma continuação dele

- Estudar a programação paralela aplicada à algoritmos de ordenação
- Implementar uma solução no modelo MapReduce, com o software Hadoop

- Comparar e analisar o desempenho das soluções com relação à quantidade de dados a serem ordenados, variabilidade dos dados de entrada e número máquinas utilizadas.

1.4 Resultados esperados

Com a realização do trabalho, busca-se ampliar e consolidar conhecimentos adquiridos na área de computação paralela, assim como a capacidade de análise e desenvolvimento de algoritmos paralelos no modelo MapReduce.

Ao final do trabalho, espera-se obter a implementação de algoritmos de ordenação paralela em ambiente Hadoop e uma análise comparativa de desempenho entre tais algoritmos.

1.5 Metodologia

O início do projeto será destinado ao estudo mais detalhado da computação paralela, em especial os algoritmos de ordenação paralela, dos fatores que influenciam o desempenho de tais algoritmos, o modelo MapReduce e a plataforma Hadoop. O passo seguinte é conhecer detalhadamente o algoritmo paralelo a ser implementado e definir as estratégias para sua implentação ambiente Hadoop. O algoritmo implementado deve ser cuidadosamente avaliado para verificar um funcionamento adequado com diferentes entradas e número de máquinas. Em seguida, serão realizados experimentos para testes de desempenho dos algoritmos com relação à quantidade de máquinas, quantidade de dados e conjunto de dados. Os resultados obtidos serão analisados e permitirão comparar a performance dos algoritmos em cada situação.

1.6 Infraestrutura necessária

A infra estrutura necessária ao desenvolvimento do projeto será fornecida pelo Laboratório de Redes e Sistemas (LABORES) do Departamento de Computação (DECOM). Esse laboratório possui um cluster formado por cinco máquinas Dell Optiplex 380, que serão utilizadas na realização dos testes dos algoritmos. Os algo-

ritmos serão desenvolvidos em linguagem Java, de acordo com o modelo MapReduce, no ambiente Hadoop.

O cluster a ser utilizado apresenta as seguintes características:

- 5 nodos
- Processador Intel Core 2 Duo de 3.0 GHz
- Disco rígido SATA de 500 GB 7200 RPM
- Memória RAM de 4 GB
- Placa de rede Gigabit Ethernet
- Sistema operacional Linux Ubunbu 10.04 32 bits (kernel 2.6.**XX**)
- Sun Java JDK 1.6.0 19.0-b09
- Hadoop 0.20.2

1.7 Cronograma de trabalho

Na Tabela **??** está descrito o cronograma esperado para o desenvolvimento do projeto. Cada atividade foi descrita para se adequar da melhor maneira ao tempo disponível do projeto, mas é possível que o cronograma seja refinado posteriormente, com a inclusão de novas atividades ou redistribuição das tarefas existentes.

Atividade	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
1	•	•								
2		•	•							
3			•							
4			•	•						
5				•	•					
6						•	•			
7								•		
8									•	
9										•

Tabela 1.1: Cronograma proposto para o projeto

1. Pesquisa bibliográfica sobre o tema do projeto e escrita da proposta
2. Estudo mais detalhado dos algoritmos de ordenação paralela, modelo MapReduce e Hadoop.
3. Configuração do ambiente Hadoop no laboratório.
4. Implementação e testes.
5. Escrita, revisão e entrega do relatório.
6. Implementação e testes.
7. Análise comparativa entre os resultados.
8. Escrita e revisão do projeto final.
9. Entrega e apresentação.

Referências

ASANOVIC, K. et al. A view of the parallel computing landscape. *Commun. ACM*, ACM, New York, NY, USA, v. 52, n. 10, p. 56–67, out. 2009. ISSN 0001-0782.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, ACM, New York, NY, USA, v. 51, n. 1, p. 107–113, jan. 2008. ISSN 0001-0782.

KALE, V.; SOLOMONIK, E. Parallel sorting pattern. In: *Proceedings of the 2010 Workshop on Parallel Programming Patterns*. New York, NY, USA: ACM, 2010. (ParaPLoP '10), p. 10:1–10:12. ISBN 978-1-4503-0127-5.

PAGE, L. et al. *The PageRank Citation Ranking: Bringing Order to the Web*. 1999.

WHITE, T. *Hadoop: The Definitive Guide*. first edition. O'Reilly, 2009. Disponível em: <http://oreilly.com/catalog/9780596521981>.