MR-Unaware para Computação Intensiva em um Ambiente Desktop Grid

Julio C. S. dos Anjos[1], **Luciana B. Arantes**[2], **Cláudio F. R. Geyer**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie – Paris – França

{jcsanjos,geyer}@inf.ufrgs.br, luciana.arantes@lip6.fr

1. Introdução

O *MapReduce*, inspirado em linguagens funcionais de alto nível, é um *framework* de programação que abstrai a complexidade do paralelismo das aplicações que tratam grande volume de dados ao oferecer as primitivas *map* e *reduce*. A arquitetura é baseada em um modelo *master/slave*. O *MapReduce* foi proposto pelo Google em 2004 para resolver problemas de busca de índice reverso em *sites* da Web e hoje é utilizado por empresas como Yahoo, Amazon, FaceBook e IBM, como infra-estrutura para *Cloud Computing*. O modelo trata a entrada de dados como uma função de *tuplas* (chave,valor), possui três fases distintas, uma de mapeamento chamada *Map* (gera dados intermediários da entrada de dados) e uma de redução chamada *Reduce* (agrupa chaves iguais geradas pela etapa de mapeamento) que são acessíveis ao programador. A terceira, é criada pelo sistema para sincronizar as duas fases e aplicar um *merge-sort* nos dados para sua transferência chamada de *Shuffle* [Dean and Ghemawat 2004]. Um gerenciamento complexo e um sistema de arquivos distribuídos e tolerante a falhas são utilizados para a execução de tarefas e garantir a integridade dos dados.

2. Motivação

O *MapReduce* foi construído para para ser utilizado em grandes *clusters* homogêneos e apresenta baixo desempenho em ambientes heterogêneos e voláteis. As *grids* de computação voluntária, como *desktop grids* [Kondo et al. 2009], são formadas por máquinas que utilizam *ciclos de idle* (ciclos ociosos de máquina) para processar as tarefas. Estas *grids* caracterizam-se por serem heterogêneas, largamente distribuídas e de alta volatilidade. A aplicação do *MapReduce* em *grid* (ao contrário de outros modelos com muitas trocas de mensagens) é viável uma vez que abstrai a paralelização de tarefas do programador [Anjos et al. 2010].

3. Trabalhos Relacionados

No trabalho de Zaharia [Zaharia et al. 2008] são abordados problemas de desempenho do *MapReduce* em ambientes heterogêneos, como em *data centers* virtualizados, originados das simplificações feitas para *clusters* homogêneos. O autor propôs um algoritmo, chamado LATE (*Longest Approximate Time to End*) para priorizar a execução de tarefas especulativas, selecionar os nós mais rápidos e evitar a degradação do sistema com tarefas especulativas. Chen [Chen and Schlosser 2008] observa que o *MapReduce* é intensivo em dados e, na implementação atual, existem problemas de desempenho como e.g. simulações físicas e processamento de dados digitais. Os trabalhos de Zaharia e Chen tratam da heterogeneidade, mas não propõem uma solução considerando a disponibilidade dos nós. No trabalho de [Lin et al. 2009] os autores propõem um modelo híbrido

de computação voluntária com o uso de máquinas não voláteis, mas não apresentam uma solução para o escalonamento de tarefas. Um algoritmo de distribuição de dados baseado na capacidade de computação de cada nó para ambientes heterogêneos foi proposto por [Xie et al. 2010]. Entretanto, a abordagem elimina o esquema da replicação de dados existente no *MapReduce*, proporcionando a degradação do sistema em caso de falhas.

4. Proposta

O Hadoop é uma implementação *open source* do *MapReduce* e será a base para este trabalho. Um sistema de arquivos distribuídos (HDFS), tolerante à falhas, garante a integridade dos dados. O objetivo desta proposta consiste em adaptar o *MapReduce* (especificamente o Hadoop) para ambientes *desktop grid*, denominado *MR-Unaware*. Serão adaptados tanto o escalonador e a distribuição inicial de dados, como criados mecanismos para garantir que a tolerância a falhas do HDFS esteja sempre disponível neste ambiente.

Um conjunto de parâmetros (C_j, D_j, t_e) são associados a cada máquina M_j para controlar o escalonamento de tarefas e distribuição de dados, onde t_e é o tempo médio das execuções de um *chunk* (menor entrada de dados) na máquina mais lenta, D_j é o tempo alocado para a computação na j-ésima máquina (obtido através de *benchmarks* e testes com a entrada de dados) e C_j é o número de *chunks* processados dentro de um tempo de execução t_e.

Os problemas a serem endereçados nesta proposta serão a distribuição de tarefas através do uso de preditor, utilização eficaz de recursos (com alterações no algoritmo de lançamento especulativo de tarefas), tratamento da volatilidade considerando a predição da disponibilidade, alocação de tarefas (map/reduce) e distribuição dos dados.

Referências

Anjos, J. C. S., Kolberg, W., Arantes, L., and Geyer, C. F. R. (2010). Estratégias para uso de mapreduce em ambientes heterogêneos. In on High Performance Computing, L. A. C., editor, *CLCAR Conferência Latinoamericana de Computación de Alto Rendimiento*, volume 1, pages 322–325. Evangraf.

Chen, S. and Schlosser, S. W. (2008). Map-reduce meets wider varieties of applications. Technical Report IRP-TR-08-05, Intel Research Pittsburgh.

Dean, J. and Ghemawat, S. (2004). Mapreduce - simplified data processing on large clusters. In *OSDI*, pages 137–150.

Kondo, D., Javadi, B., Malecot, P., Cappello, F., and Anderson, D. P. (2009). Cost-benefit analysis of cloud computing versus desktop grids. In *IPDPS '09 - Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, pages 1–12, Washington, DC, USA. IEEE Computer Society.

Lin, H., Archuleta, J., Ma, X., chun Feng, W., Zhang, Z., and Gardner, M. (2009). Moon - mapreduce on opportunistic environments. Technical report, Virginia Polytechnic Institute and State University.

Xie, J., Yin, S., Ruan, X., Ding, Z., Tian, Y., Majors, J., Manzanares, A., and Qin, X. (2010). Improving mapreduce performance through data placement in heterogeneous hadoop clusters. In *2010 IEEE International Symposium on Parallel & Distributed Processing*, Workshops and Phd Forum (IPDPSW), pages 1–9.

Zaharia, M., Konwinski, A., Joseph, A. D., Katz, Y., and Stoica, I. (2008). Improving mapreduce performance in heterogeneous environments. *OSDI*.