

# Estudo de Viabilidade do Uso de MapReduce em Ambiente Desktop Grid

Julio C. S. Anjos, Cláudio R. Geyer

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{jcsanjos,geyer}@inf.ufrgs.br

## 1. Introdução

Estruturas de dados da ordem de Petabytes fomentam a necessidade de se encontrar alternativas que viabilizem o tratamento dos dados através do uso de sistemas de computação intensivos. Estes sistemas não privilegiam somente o processamento mas também a maneira com que os dados devem ser manipulados. Este trabalho encontra-se na fase de levantamento de dados e investiga a viabilidade do uso de uma estrutura *MapReduce* [White 2009] aplicada a ambientes heterogêneos como em computação voluntária [Buyya et al. 2000], para abstrair o paralelismo do programador e tratar grandes volumes de dados, agregando tolerância a falhas de maneira intrínseca.

## 2. Trabalhos Relacionados

Alguns estudos e implementações utilizando o *MapReduce* foram feitos para o tratamento de grandes volumes de dados [Miceli et al. 2009], sendo usado como um framework de programação que abstrai a complexidade do paralelismo das aplicações. Esta técnica apresenta um desempenho considerável porque seu escalonador pressupõe que haja a homogeneidade dos nós do cluster, tendo um mecanismo simples de tolerância a falhas, confia que suas tarefas serão executadas linearmente. Entretanto este modelo é projetado para ambientes homogêneos e não é adequado à heterogeneidade existente em grids de computação voluntária, apresentando um baixo desempenho [Zaharia et al. 2009]. Outros estudos recentes [Buyya et al. 2009], [Foster et al. 2008], procuram definir o tratamento deste tipo de dados em uma arquitetura *Cloud Computing* diferenciando-se dos sistemas computacionais existentes.

## 3. Descrição do Modelo

O *MapReduce* apresenta três fases distintas, *Map* (gera *tuplas* com [chave,valor] da entrada de dados), *shuffle* (fase de transferência e armazenamento intermediária de dados) e *Reduce* (gera nova chave com o agrupamento de *tuplas* [chave,valor] iguais). Uma fase *Reduce* só inicia depois que todos os dados intermediários tenham sido gerados, ocorrendo falhas em um *Map* o sistema gera novamente aquele dado perdido, degradando todo o sistema.

Em função das aplicações tenderem a iniciar e terminar rapidamente, o modelo do escalonador admite algumas simplificações considerando que os nós executam tarefas de tamanho semelhante. Com tempo de execução igual em todas as fases, o progresso de tarefas ocorreria em frações constantes de 1/3 do tempo total de processamento. Baseando-se nisto, o escalonador confia que as tarefas seriam homogêneas e iriam requerer o mesmo tempo de execução entre elas. Assim para minimizar o tempo de resposta,

o *MapReduce* roda no modo especulativo, fazendo a cópia da tarefa que está rodando em um nodo lento, para ser executada em outro mais rápido, o que não causaria custo a priori [Zaharia et al. 2009]. Estas premissas são válidas em alguns aspectos para ambientes de *clusters* homogêneos. Em ambientes heterogêneos como em *Desktop Grid*, com um comportamento distinto, tornam-se inadequadas.

O modelo avaliado constitui-se de uma estrutura mestre/escravo, onde o mestre é o controlador e os escravos são estações que compõem a rede de maneira voluntária, semelhante à estrutura da Grid XtremWeb [Kondo et al. 2007]. Parte do estudo é feito sobre a plataforma do *Hadoop* que é uma implementação do *MapReduce* diferentemente do modelo original, as tarefas tem blocos de tamanho variável conforme a disponibilidade dos recursos computacionais. No modelo proposto, os nodos trabalhadores (*workers*) informam sua disponibilidade ao mestre que distribui as tarefas a estes *workers*. Agregando-se um preditor para a detecção de *crash* nos nodos, com uso de algoritmos de escalonamento estocástico [Javadi et al. 2009], minimiza-se a degradação do sistema.

#### 4. Resultados Esperados

Como resultados do estudo, busca-se aumentar o desempenho do *MapReduce*, investigando-se o uso de algoritmos de escalonamento estocástico para aplicação em ambientes heterogêneos e permitir a computação intensiva em *desktop grid* com desempenho aceitável, criando-se mecanismos para o tratamento de informações de grande escala.

#### Referências

- Buyya, R., Baker, M., and Laforenza, D. (2000). The grid: International efforts in global computing. Technical Report 1, Grid Computing Info Centre.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, (25):599–616.
- Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10.
- Javadi, B., Kondo, D., Vincent, J., and Anderson, D. (2009). Mining for statistical availability models in large-scale distributed systems: An empirical study of seti@home. In *17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*.
- Kondo, D., Fedak, G., Cappello, F., Chien, A. A., and Casanova, H. (2007). Characterizing resource availability in enterprise desktop grids. *Future Generation Computer Systems*, 23:888–903.
- Miceli, C., Miceli, M., Jha, S., Kaiser, H., and Merzky, A. (2009). Programming abstractions for data intensive computing on clouds and grids. *9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009. CCGRID '09*, 1(9):478–483.
- White, T. (2009). *Hadoop: The Definitive Guide*, volume 1. O'Reilly Media, Inc., 1 edition.
- Zaharia, M., Konwinski, A., Joseph, A. D., Katz, R., and Stoica, I. (2009). Improving mapreduce performance in heterogeneous environments.