Suportando Mineração de Dados em Arquiteturas de Alto Desempenho

Élder F. F. Bernardi[1], **César A. F. De Rose**[1]

[1]Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681 - Prédio 32 – Porto Alegre – RS – Brasil

{elder.bernardi,cesar.derose}@pucrs.br

1. Introdução

Uma das áreas que buscam nas arquiteturas de alto desempenho uma base para fornecimento de poder computacional para a execução de suas aplicações é a área de Mineração de Dados. Tarefas de Mineração de Dados consistem na aplicação de algoritmos sobre grandes bases de dados a fim de se extrair conhecimento não previsível e não trivial [Tan et al. 2006]. A execução de tais tarefas demanda grande poder computacional e um volume considerável de manipulação e transmissão de dados, o que torna essas aplicações altamente propícias a serem executadas em ambientes de alto desempenho.

Embora já existam ferramentas que suportem a execução de tarefas de mineração nesses ambientes [Teodoro et al. 2008], [Talia et al. 2008], essas ferramentas demandam que as aplicações a serem nelas executadas utilizem um único modelo de programação e sejam desenvolvidas através de ferramentas de programação específicas para a ferramenta. Além disso, demandam a existência de camadas de software que implicam na alteração da infraestrutura dos ambientes de alto desempenho de seus usuários. Esses requisitos tornam essas ferramentas restritas a usuários com conhecimento avançado em computação de alto desempenho, afastando usuários que de fato são especialistas em mineração de dados mas não possuem conhecimento avançado em gerência de ambientes de alto desempenho.

Diante desta contextualização, acrescentando-se o papel estratégico que aplicações de mineração desempenham dentro de diversas organizações, propõe-se uma arquitetura que tem como propósito fornecer ferramentas de fácil utilização que permitam a execução de aplicações de mineração de dados paralelas em ambientes de alto desempenho.

2. Arquitetura Proposta

A arquitetura proposta apresenta funcionalidades que permitem a utilização de recursos computacionais de forma transparente ao usuário, sem demandar conhecimentos técnicos para o uso desses. Para isso, construiu-se uma organização possuidora de prospectadores de recursos capazes de reconhecer e fazer previsões sobre a capacidade dos recursos fornecidos pelo usuário. Através dessa prospecção, são colhidas informações que servem de apoio aos escalonadores da arquitetura. Esses são responsáveis por decompor a execução da aplicação em tarefas menores e mapeá-las nos recursos computacionais disponíveis. Uma vez mapeadas as tarefas, a arquitetura se encarrega de realizar a execução e monitoração automáticas dessas tarefas. Todo esse processo ocorre sem necessitar a instalação de softwares de fins exclusivos para a arquitetura nos recursos a serem utilizados.

A arquitetura suporta dois níveis de paralelismo em aplicações de mineração: paralelismo de dados e paralelismo de fluxo de execução sobre esses dados. Em termos de suporte de recursos, suporta arquiteturas do tipo SMP, cluster e ambientes de grade. O escalonamento de recursos foi desenvolvido com ênfase no aproveitamento de processadores multinúcleos presentes nos recursos, possibilitando uma abordagem de paralelização e escalonamento híbrido (nodos X núcleos de processamento).

Outro aspecto considerado pela arquitetura é a gerência e manipulação dos dados exigidos pelas tarefas de mineração. Nesse sentido, construí-se mecanismos que auxiliam o acesso a dados a partir dos recursos computacionais onde as tarefas estão em execução. Criou-se mecanismos que capacitam as tarefas remotas a acessarem seus dados da maneira mais eficaz, de acordo com o tipo de recurso ao qual ela está alocada e a localização dos dados a serem acessados. Para tal, criou-se um gerenciador de dados que é responsável pelo armazenamento das bases de dados a serem utilizadas e provê acesso a esses via acesso direto em disco, cópia remota ou acesso Web por meio de Web Services.

A fim de validar o ambiente desenvolvido, paralelizou-se um algoritmo de mineração de dados para regressão e se realizou a execução desse através da arquitetura, explorando tanto o paralelismo de dados tanto quanto o de fluxo de execução.

3. Resultados Obtidos e Trabalhos Futuros

Entre os resultados obtidos, destaca-se:

- Criação de uma arquitetura que gerencia desde a aquisição e adaptação de dados para tarefas de mineração, até o mapeamento e execução de aplicações de mineração paralelas nos recursos fornecidos pelo usuário.
- Criação de um algoritmo de escalonamento capaz de mapear tarefas de tamanhos dinâmicos a recursos computacionais heterogêneos sem ter conhecimento prévio sobre o comportamento dessas tarefas nos recursos, que obteve um desempenho aceitavelmente próximo do ótimo. Salienta-se também a capacidade desse escalonador tomar proveito de arquiteturas multinúcleos, facilitando o escalonamento de aplicações híbridas, como MPI com OpenMP, por exemplo.
- Paralelização de uma aplicação para regressão com potencial de aceleração de 42 vezes em relação a sua versão sequencial, utilizando-se 10 nodos computacionais com seis unidades de processamento cada.

Futuramente pretende-se incluir o fator localidade de dados para tomada de decisão no mapeamento de tarefas a recursos.

Referências

Talia, D., Trunfio, P., and Verta, O. (2008). The weka4ws framework for distributed data mining in service-oriented grids. *Concurrency and Computation: Practice and Experience*, 20(16):1933–1951.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Teodoro, G., Fireman, D., Guedes, D., Meira, W., and Ferreira, R. (2008). Achieving multi-level parallelism in the filter-labeled stream programming model. In *ICPP '08: Proceedings of the 2008 37th International Conference on Parallel Processing*, pages 287–294, Washington, DC, USA. IEEE Computer Society.