

Middleware Clarens no Contexto da Física de Altas Energias

Diego S. Gomes, Marko Petek,
Tiarajú A. Diverio, Cláudio F. R. Geyer

PPGC, Instituto de Informática, UFRGS
Porto Alegre - RS
{dsgomes, petek, diverio, geyer}@inf.ufrgs.br

22 de novembro de 2005

1 Introdução

Os diversos experimentos da área da Física de Altas Energias (HEP - *High Energy Physics*) geram grandes quantidades de dados (na ordem de petabytes de dados – $1 PB = 10^{15} bytes$). Estima-se, com a operacionalidade de novos projetos até a próxima década, que estes conjuntos de dados aumentem em tamanho para a ordem de Exabytes [4].

Neste cenário, há necessidade de uma estrutura computacional capaz de prover acesso, processamento e análise desta grande quantidade de dados. Aplica-se então o uso de grades computacionais, de forma que os diversos grupos de pesquisa de HEP possam colaborar. Dentre estes grupos, pode-se citar o de Caltech, que possui uma cooperação com o grupo GPPD [2] da UFRGS.

Dentre os middlewares desenvolvidos por Caltech, existe o Clarens [1]. O Clarens é um *framework* para grades computacionais baseado em Web Services. Seu objetivo principal é fornecer uma estrutura capaz de facilitar a busca e análise da imensa quantidade de dados que serão gerados pelo experimento CMS no LHC [3], que está sendo construído no CERN.

Este trabalho tem o objetivo de testar alguns dos mecanismos de comunicação fornecidos pelo Clarens, de forma a verificar sua aplicabilidade em soluções propostas pelo grupo para o problema da busca e armazenamento dessa grande quantidade de dados. Como contribuição, os resultados de algumas das medições são apresentados.

2 Testes e Resultados

A máquina onde os testes foram feitos é um Athlon XP 2400+ com 512Mb de memória, rodando a distribuição Rocks Cluster 3.3.0 e o Clarens versão 0.7.0-1.

Para cada método, mediu-se o tempo para fazer-se 100 chamadas. Os testes foram efetuados em um horário do dia onde o uso dos recursos da máquina era baixíssimo por parte de outros serviços, e é possível visualizá-los na Tab.1. O teste #1 mediu o tempo de instanciação de um objeto cliente Clarens e ficou na média de 0.5569s/chamada. Em #2 programou-se uma chamada para *echo* sem argumentos (ou seja, o método retornava

#	Método	Total em seg. (100 chamadas.)
1	<code>Clarens.client()</code>	55.692157
2	<code>teste.echo()</code>	4.672356
3	<code>teste.echo(strgrande)</code>	5.182908
4	<code>teste.shellldata()</code>	6.631747
5	<code>file.read_tar(['\'/'\''])</code>	60.062515
6	<code>system.logout()</code>	56.746905

Tabela 1: Resultados das medições

uma string vazia). Em contrapartida, tentou-se esconder o *overhead* do XML invocando-se `echo` com uma string grande (295 caracteres), mas, como pode ser visto na tabela, a diferença foi pequena. `shellldata` (teste #4) teve um tempo de execução ainda maior, mas desta vez devido ao tempo gasto na computação de `/bin/date`. #5 objetivou testar o método `read_tar` do serviço `file` que acompanha o Clarens. Este método retorna um binário, que é um arquivo `.tar` do diretório raiz Clarens (ou seja, é um pacote contendo todos os arquivos visíveis aos clientes - aproximadamente 500Kb). Por fim, #6 dá o tempo de `logout`, surpreendentemente alto quando comparado ao tempo de instanciação.

Cada medição de 100 chamadas foi executada diversas vezes (mais de 20), e a tabela mostrada apresenta o resultado da medição com o **menor** tempo. Tal abordagem foi adotada devido ao uso concorrente de outros recursos de Gradep (pois esta concorrência eleva os tempos de execução dos serviços do Clarens).

3 Conclusões e Trabalhos Futuros

Ainda não é possível concluir quanto ao desempenho do Clarens pois precisa-se medir o impacto dos mecanismos de comunicação de outros sistemas para obter um valor comparativo. Assim, pretende-se fazer testes semelhantes com sockets, RMI e outras bibliotecas para web services (ex.: PEAR), bem como implementá-los no toolkit do Globus.

Como trabalho relacionado, [1] apresenta medições quanto ao uso simultâneo do *middleware* por parte de diferentes clientes simultâneos, mostrando boa escalabilidade.

Referências

- [1] Clarens. World Wide Web. <http://clarens.sourceforge.net/index.php?intro>, consultado em Outubro de 2005.
- [2] Grupo de Processamento Paralelo e Distribuído. World Wide Web. <http://gppd.inf.ufrgs.br/>, consultado em Outubro 2005.
- [3] The Large Hadron Collider. World Wide Web. <http://www.cern.ch/LHC>, consultado em Setembro de 2005.
- [4] Julian J. Bunn and Harvey B. Newman. Data intensive grids for high energy physics. In Fran Berman, Geoffrey Fox, and Tony Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*, chapter 39. Wiley, New York, March 2003.