

Um Protótipo Para Visualização Científica Remota com o Cluster GradeBR/UFAL

Marco Antonio de A. Silva^{1,2}, Baltazar T. Vanderlei^{1,3}, Leonardo P. Viana^{1,2,3}

¹Instituto de Computação – IC
Universidade Federal de Alagoas – UFAL

²Centro de Pesquisa em Matemática Computacional – CPMAT
Universidade Federal de Alagoas – UFAL

³Laboratório de Computação Científica e Visualização – LCCV
Universidade Federal de Alagoas – UFAL
BR 104 Norte km 97, 57072-970 Maceió, AL

marcoantonio.maas@gmail.com

psycho@lccv.ufal.br, lpviana@lccv.ufal.br

Resumo. *Este artigo apresenta uma proposta para a visualização científica remota de grandes volumes de dados. Esta solução faz uso das Placas Gráficas de Propósito Geral (GPGPUs) presentes no cluster. A implementação está em fase inicial, procuramos minimizar os custos em termos de rede e processamento do lado cliente. Somente foram adotadas tecnologias multiplataformas e de código aberto.*

1. Introdução

Um dos problemas centrais da computação científica é a geração de uma enorme massa de dados. Através de simulações ou experiências são obtidos geralmente um grande volume de dados numéricos. Quando são utilizadas técnicas de modelagem computacional e sensoriamento de fenômenos dinâmicos em sistemas complexos naturais ou artificiais, aliadas a técnicas de computação de alto desempenho, o volume de dados a ser tratado nestes problemas atualmente pode ser da ordem de petabytes [Wu et al. 2009]. A transformação de dados em forma visual, usualmente 3D com componente dinâmico (tempo ou animação), torna sua análise humanamente factível. Analisar, tirar as informações relevantes, destes dados, muitas vezes associados a fenômenos dinâmicos em três ou mais dimensões, faz parte do escopo da *visualização científica* [Friendly and Denis 2001]. Em projetos de larga escala a armazenagem, transferência e organização destes dados, por si só, se tornam problemas de difícil solução.

O modelo apresentado tem como objetivo tornar viável a visualização remota de grandes volumes de dados gerados em modelagem computacional e simulação no cluster GradeBR/UFAL do Laboratório de Computação Científica e Visualização, fazendo uso de suas Placas Gráficas de Propósito Geral (GPGPU) e de seu sistema de arquivos paralelo de alta capacidade. Um dos problemas iniciais tratado, foi o de minimizar ao máximo o tráfego de dados pela rede. Para poder utilizar o processamento das GPGPUs é necessário gerar visualização dos dados localmente e depois enviar as imagens para o cliente. Isso

foi feito com a configuração de um servidor de visualização remota e um software que permitisse que a renderização do OpenGL fosse feita no servidor, enviando apenas as imagens ao cliente, os invés do conjunto completo de dados.

2. Visualização Científica Remota usando Cluster Computacional

A Petrobras financiou, através da Rede de Computação Científica e Visualização – Rede Galileu, a criação do cluster GradeBR/UFAL no Laboratório de Computação Científica e Visualização (LCCV). Uma das metas desta Rede é criar um ambiente de grid de alto desempenho continental, a GradeBR, com desempenho teórico máximo (Rpeak) de 160 Teraflops, capaz de fornecer com eficiência capacidade de processamento para problemas de engenharia computacional aplicados à exploração de óleo e gás. Um dos objetivos também é criar um sistema em teraescala de visualização 3D em tempo real para problemas de mecânica computacional aplicada [Oliveira et al. 2010].

2.1. Cluster

O *cluster* GradeBR-UFAL, instalado nas dependências do LCCV, é uma das partes que compõe a maior grade de computadores da América Latina, possui a função de realizar cálculos e processar os dados necessários a exploração de petróleo na área do Pré-Sal. Um resumo da capacidade computacional do cluster GradeBR é apresentado na Tabela 1.

Tabela 1. Informações sobre o cluster [Vanderlei and Pereira 2010]

INFORMAÇÃO	DADOS
Nome	GradeBR-UFAL
Tipo	Hípercubo 4D
Tecnologia	Sun Fire 2X2S
Processadores (CPUs)	Intel Xeon X5560/W5580 2,8/3,2 GHz de 4 núcleos
Processadores Gráficos de Uso Geral (GPGPU)	nVidia Tesla S1070
Tamanho	1.752 núcleos de processamento em CPU (1496 Intel Nehalem 2,8 GHz, 256 Intel Nehalem 3,2 GHz) e 7.680 núcleos de processamento em GPGPU (Nvidia cores)
Rede	Infiniband QDR (40 Gbps)
Memória	3 GB de RAM por núcleo de CPU (total de 5,256 TB)
Armazenamento	90 TB em disco (50 TB LustreFS + 40 TB NFS)
Desempenho máximo teórico (Rpeak)	20,07 Tflops em CPUs + 32 Tflops em GPGPU com precisão simples

2.1.1. Arquitetura de Rede InfiniBand

InfiniBand (IB) é uma arquitetura ponto-a-ponto que define um canal de entrada e saída (I/O Channel) de alta velocidade para interligar servidores, equipamentos de infraestrutura de comunicação, sistemas de armazenamento e embarcados [de Melo Carvalho Filho et al. 2010]. Por oferecer baixa latência, elevada largura de banda e carga de CPU extremamente baixa com uma ótima relação custo/benefício, o InfiniBand se tornou

a mais implantada interconexão de alta velocidade [Pfister 2001]. Atualmente, Infini-Band fornece throughput de até 40 Gb/s servidor para o servidor e 120Gb/s switch para switch. Essa largura de banda de alto desempenho é aliada com a baixíssima latência, e este desempenho tem quase custo zero em termos de utilização da CPU. [Grun 2010]

2.1.2. Lustre

Na procura de um sistema de arquivos que possibilite um maior desempenho, aumentando o throughput (taxa de transferência) de dados e garantindo que esses dados possam ser os mesmos para uma grande quantidade de processos ou usuários, foram encontrados sistemas de arquivo que provêm maior desempenho. Dentre esses sistemas, se destaca o Lustre [da Silva et al. 2010]. O Lustre é um sistema de arquivos paralelo distribuído de código aberto desenvolvido e mantido pela Sun Microsystems Inc. (Oracle). Devido à arquitetura extremamente escalável do sistema de arquivos Lustre, o seu uso é popular nas aplicações de computação científica, bem como na indústria de óleo e gás, manufatura, mídia e no setor financeiro. O Lustre apresenta uma interface POSIX aos seus clientes com capacidade de acesso paralelo. Boa parte dos supercomputadores mais rápidos em todo o mundo usam o sistema de arquivos Lustre para a alta desempenho [Wang et al. 2009].

2.2. Ferramentas

Para poder utilizar o processamento das GPGPUs é necessário gerar a visualização dos dados localmente e depois enviar as imagens para o cliente, para tanto pesquisou-se ferramentas com essas funcionalidades. Sendo as escolhidas: VirtualGL e TurboVNC.

2.2.1. VirtualGL

Quando uma aplicação OpenGL é executada através de um servidor remoto X, seus comandos OpenGL são transmitidos para o cliente, que ficará encarregado pelo processamento gráfico. O VirtualGL é um pacote de código aberto que permite que um Linux remoto execute aplicações OpenGL com aceleração via hardware. Com ele comandos OpenGL são renderizados no servidor e as imagens da aplicação 3D são geradas [VirtualGL 2010].

2.2.2. TurboVNC

O VNC é um sistema baseado em um simples protocolo de exibição e é independente de plataforma. Permite o uso remoto de máquinas, evitando que o usuário tenha que carregar consigo o hardware. O VNC permite que um único computador seja acessado simultaneamente a partir de vários locais, assim permitindo o trabalho cooperativo. Ao contrário de outros protocolos de exibição remota, como o *X Window System*, o protocolo VNC é totalmente independente de sistema operacional, sistema de janelas e aplicativo [Richardson et al. 2002]. O Projeto VirtualGL fornece uma versão acelerada do VNC, chamado "TurboVNC", que é feita para ser usado com o VirtualGL. A combinação dos dois oferece uma solução de alto desempenho para o 3D remoto, mesmo em redes lentas.

O TurboVNC também fornece recursos de colaboração, permitindo simultaneamente a vários utilizadores interagir com o mesmo aplicativo 3D [VirtualGL 2010].

2.3. Pipeline

Em nosso modelo, como pode ser visto na Figura 1, o armazenamento (LustreFS) e o servidor de renderização estão no mesmo ambiente e se comunicam através da rede InfiniBand, ou seja não há transferência externa de dados. O servidor de renderização se utiliza de um servidor VNC (aplicação) combinado com o VirtualGL para enviar somente as imagens geradas pela aplicação 3D para o(s) cliente(s), além disso o VNC permite a interação entre o cliente e a tal aplicação.

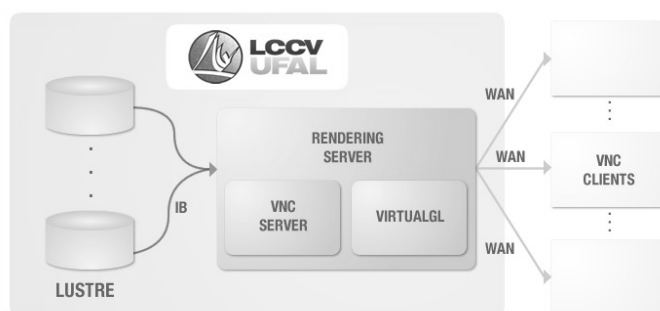


Figura 1. Pipeline proposto, o armazenamento e o servidor de renderização estão no mesmo ambiente e se comunicam através da rede InfiniBand.

2.4. Chromium

Estamos usando o *Chromium*, em fase experimental, para fazer a decomposição das imagens a serem processadas em paralelo, assim melhorando a taxa de quadros por segundo e a resolução, além de poder renderizar um conjunto maior de dados.

O Chromium é um sistema de manipulação do fluxo de comandos da API gráfica. Ele possui vários filtros, entre eles destacam-se os que, podem ser arranjados para distribuir a visualização e os que paralelizam o processamento gráfico. Esses filtros podem ser facilmente estendidos e personalizados utilizando-se scripts escritos em Python, permitindo ao usuário determinar o fluxo de processamento realizado em cada nó. O Chromium alcança boa escalabilidade em clusters, além de abstrair completamente a arquitetura do processamento gráfico, topologia de rede, e a semântica de processamento dos comandos da API [Humphreys et al. 2002].

3. Resultados

Atualmente estamos analisando e fazendo testes com o Chromium, ao paralelizar a renderização para obter ganhos de quadros por segundo. Também estamos analisando uma forma de disponibilizar um acesso prático aos servidores de renderização, mas sem esquecer da segurança. Estamos trabalhando em uma forma de disponibilizar imagens 3D estereoscópica, alguns testes já foram executados com êxito, mas a configuração no lado cliente não era muito amigável.

O sistema ainda está em fase de prototipação, havendo instabilidade, pode ser acessado através de qualquer conexão de banda larga e por dispositivos com pouco poder de processamento, como *tablets* e celulares.

Agradecimento

Agradecemos a ANP, a Petrobras e ao Laboratório de Computação Científica e Visualização da Universidade Federal de Alagoas por garantir acesso aos recursos computacionais do cluster GradeBR/UFAL da Rede Galileu.

Referências

- da Silva, A. S., Vanderlei, B. T., Leite, J. B., and Pereira, L. V. (2010). Sistema paralelo de arquivos em rede de alto desempenho no cluster gradebr/ufal. *Anais do VII Congresso Acadêmico da Universidade Federal de Alagoas*.
- de Melo Carvalho Filho, A., Rego, D. C. G., Pereira, L. V., and Vanderlei, B. T. (2010). Arquitetura infiniband: características e estudo de caso do cluster gradebr/ufal. *Anais do VII Congresso Acadêmico da Universidade Federal de Alagoas*.
- Friendly, M. and Denis, D. (2001). Milestones in the history of thematic cartography, statistical graphics, and data visualization. *documento web, disponível em <http://www.math.yorku.ca/SCS/Gallery/milestone>*, 174.
- Grun, P. (2010). Introduction to infiniband for end users. *documento web, disponível em http://members.infinibandta.org/kwspub/Intro_to_IB_for_End_Users.pdf*.
- Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P., and Klosowski, J. (2002). Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Transactions on Graphics*, 21(3):693–702.
- Oliveira, A. L. C., Vanderlei, B. T., and Pereira, L. V. (2010). Infraestrutura de rede para o grid continental gradebr: Proposta para o nó gradebr/ufal. *Anais do VII Congresso Acadêmico da Universidade Federal de Alagoas*.
- Pfister, G. (2001). An introduction to the infiniband architecture. *High Performance Mass Storage and Parallel I/O*, pages 617–632.
- Richardson, T., Stafford-Fraser, Q., Wood, K., and Hopper, A. (2002). Virtual network computing. *Internet Computing, IEEE*, 2(1):33–38.
- Vanderlei, B. T. and Pereira, L. V. (2010). Implementação de um supercomputador: cluster gradebr/ufal. *Anais do VII Congresso Acadêmico da Universidade Federal de Alagoas*.
- VirtualGL (2010). User's guide for virtualgl. *documento web, disponível em http://www.virtualgl.org/vgldoc/2_1_4final/*.
- Wang, F., Oral, S., Shipman, G., Drokin, O., Wang, T., and Huang, I. (2009). Understanding lustre filesystem internals.
- Wu, Q., Gao, J., Chen, Z., and Zhu, M. (2009). Pipelining parallel image compositing and delivery for efficient remote visualization. *Journal of Parallel and Distributed Computing*, 69(3):230–238.