

DISTRIBUIÇÃO DE CLUSTER PARA ANÁLISE DE PERFORMANCE

Paulo R. Gifalli¹

¹Centro de Ciências Exatas e Sociais Aplicadas – Universidade do Sagrado Coração
Rua Irmã Arminda – 10-50 – 17011-160 – Jardim Brasil, Bauru – SP – Brasil

gifallikraft@gmail.com

Abstract. *Increasingly, current applications require high processing power, such as the game development, application of special effects to a scene in a movie. Based on this need for computational power, is studying some tools for image rendering. In this work, an evaluation of the performance rendering tools for pre-formatted images making use of clusters, consisting of computers have multicore architectures That. In the Theoretical study done with the affairs belonging to the subject, we Performed a case study using the Blender rendering tool. The operating system, Ubuntu Server x86 version 11, Which possesses resources of parallelism, so Can you see the division of an animation into a set of sub-tasks, Thus making it an independent process, process migration Achieving Between nodes and making use of the Nucleus present in cluster of multicore computers.*

Resumo. *Increasingly, current applications require high processing power, such as a game development, application of special effects to a scene in a movie. Based on this need for computational power, is studying some tools for image rendering. In this work, an evaluation of the performance rendering tools for pre-formatted images making use of clusters, consisting of computers that have multicore architectures. In the theoretical study done with the affairs belonging to the subject, we performed a case study using the Blender rendering tool. As operating system, Ubuntu Server x86 version 11, which possesses resources of parallelism, so you can see the division of an animation into a set of sub-tasks, thus making it an independent process, achieving process migration between nodes and making use of the cluster of nucle present in multicore computers.*

Introdução

Segundo Manika (2000), diversas empresas de pequeno porte e universidades, com poucos recursos financeiros estão recorrendo a sistemas de cluster como alternativa. Isso Permite que essas instituições possam aproveitar seus *hardwares* existentes para implementação desse sistema. Com relação às áreas de aplicações, os clusters podem ser usados em diversos ramos de atividades, como: na área de sensoriamento remoto onde necessitam de alto desempenho para processar imagens de satélites; outra área muito utilizada é a da engenharia genética, onde um cluster pode ser de grande utilidade no projeto Genoma; em exploração sísmica para determinação de local de poços petrolíferos, através de processamento rápido para simulação vetorial de uma região; na área de previsão de tempo, para realizar simulações climáticas de maneira rápida e com grande precisão, permitindo que os resultados das previsões sejam observada pelos pesquisadores com antecedência, antes da realização do fenômeno climático; uma outra importante área onde um cluster é utilizado é na segurança de reatores nucleares (monitorações críticas em tempo real), com intuito de analisar as condições dos reatores, através de simulações de operações, assim evitando possíveis acidentes.

Quando se fala em renderização de imagem fala-se em alto poder computacional, com base nesta informação, estes extremos propõe testes de capacidade computacional, por meio da renderização de imagens, hoje um campo bastante procurado, como por exemplo as empresas automobilísticas, nos desenvolvimento dos seus mais variados modelos de carros, como também na área de animações de filmes, etc (ALECRIM, 2011).

2 CLUSTER

A resolução de diversos problemas científicos e comerciais complexos exige grande esforço computacional. Computadores de grande porte têm sido utilizados com frequência nesse tipo de aplicação (YOKOKURA, 2007).

Para alcançar grande poder de processamento, uma abordagem que fornece alto desempenho computacional e processamento paralelo. Dentre as abordagens utilizadas para processamento de alto desempenho, podem ser destacadas a utilização de supercomputadores *clusters* de computadores (PITANGA, 2003).

A determinação de propriedades físicas dos materiais com uso das ferramentas da análise de imagens (GONZALES, 1993).

Segundo Yokokura (2007), os supercomputadores são extremamente caros, bem como os chamados *MPP's* (*Massively Parallel Processos*), multicomputadores construídos com milhares de processadores comerciais conectados por uma rede proprietária de alta velocidade, uma alternativa com custo muito menor e que resulta em bons resultados, ou seja, garante um bom poder de processamento são os *clusters* de computadores, com um custo muito baixo e alto desempenho, os *clusters* têm se tornado a cada dia uma alternativa aos *MPP's* (*Massively Parallel Processors*) supercomputadores.

Segundo Perry (2005), os clusters são representantes da Arquitetura *MIMD* (*Multiple Instruction Multiple Data*) são arquiteturas caracterizadas pela execução simultânea de múltiplos fluxos de instruções. Essa capacidade deve-se ao fato de que são construídas a partir de vários processadores operando de forma cooperativa ou concorrente, na execução de um ou vários aplicativos. Essa definição deixa margem para que várias topologias de máquinas paralelas e de redes de computadores sejam enquadradas como *MIMD*. A diferenciação entre as diversas topologias *MIMD* é feita pelo tipo de organização da memória principal, memória cache e rede de interconexão, com memória distribuída, onde também incluímos as *MPP's* (*Massively Parallel Processors*), enquanto nas Arquiteturas *MIMD* com memória compartilhada temos as *SMP's* (*Symmetric Multiprocessing*), todos representantes de arquiteturas paralelas.

Segundo dados da Conferência Internacional de Supercomputação realizada em Junho de 2010, dentre os 500 computadores mais poderosos do mundo, 304 (60.8 % do total) são *clusters* de computadores. O *cluster* mais potente do mundo ocupa a 1ª posição na lista dos 500 computadores mais poderosos do mundo. Trata-se do Roadrunner Cluster, localizado no *USA*.

São 13.140 processadores, em 6.562 nós dual Power com 3.200 MHz (1.02 PetaFlops), atingindo um pico de processamento de 1.02 quatrilhões por segundo, com seus 80 TB de memórias. O cluster possui sistema operacional Linux e rede de interconexão *Myrinet*, ocupando 288 *hacks* e ocupa uma área de cerca de 557,41 m² com um peso de 226,796 Kg. (DIGNOW, 2011).

O cluster foi uma iniciativa do governo americano com a *IBM*, com finalidades de utilizar na segurança nacional dos *USA*, testar material nuclear e sistemas de armas

nucleares, previsão de tempo e estudo do universo (ACIDIT, 2011).

Segundo o líder da divisão de computação de alta performance do laboratório de Los Alamos, John Morrison, a curto prazo de tempo, é montar um mapeamento do *HIV* e tenta achar a cura (DIGNOW, 2011).

Segundo Yokokura (2007), o cluster de computadores são uma alternativa aos supercomputadores, para prover alta performance e alta disponibilidade, e é particularmente atrativo para aplicações de servidores. No final de 1993, Donald Becker e Thomas Sterling, ambos cientistas da *CESDIS* (Center of Excellence in Space Data and Information Sciences) da *NASA*, iniciaram um esboço de um sistema de processamento distribuído construído a partir de hardware convencional como uma medida de combate aos custos dos supercomputadores. No início de 1994, nasce o primeiro *cluster* e o projeto Beowulf. O protótipo inicial era um cluster de 16 processadores *DX4* ligados por dois canais *Ethernet* acoplados. A máquina foi um sucesso instantâneo, e esta ideia rapidamente se espalhou pelos meios acadêmicos, pela *NASA* e outras comunidades de pesquisa. A utilização de clusters tem sido uma alternativa cada vez mais viável para suprir a demanda do crescente número de aplicações que exigem alto desempenho computacional. Têm sido largamente utilizados no meio acadêmico e comercial para a execução de aplicações ditas de “execução críticas”, geralmente, relacionadas a sistemas de tempo real ou que manipulam uma grande quantidade de dados.

Portanto é possível definir que *cluster* é formado com grupos de computadores interconectados trabalhando de forma conjunta como um único computador. Os computadores do *cluster* se comportam como sistemas que podem, opcionalmente, funcionar de forma independente do *cluster*. Cada computador do *cluster* é tipicamente referenciado como um nó (STALLINGS, 2000). Nessa arquitetura, normalmente, existe um nó mestre que gerencia ou divide as tarefas entre os demais nós, chamados de escravos (PITANGA, 2004).

2.1 GRID COMPUTING

Segundo Foster (2003) define *Grid* como uma infraestrutura emergente que irá mudar fundamentalmente a forma como pensamos e utilizamos os computadores. A palavra *Grid* é utilizada como uma analogia as grades energia, que provem acesso pervasivo a eletricidade, da mesma forma que o computador e outros pequenos avanços, proporcionaram um dramático impacto nas capacidades humanas e na sociedade.

Foster (2003) afirmando que muitos acreditam que, por permitir que todos os componentes de nossa infraestrutura de tecnologia da informação – capacidades computacionais, bancos de dados, sensores e pessoas – sejam compartilhados de uma forma flexível através de ferramentas colaborativas, *Grid* terá um efeito de transformação similar, permitindo o surgimento de novas classes de aplicação.

O impacto previsto por Foster (2003), com o surgimento dos *Grids* não se concretizou comercialmente. Por outro lado, a tecnologia desenvolvida para os *Grids* evoluiu e vem sendo utilizada nos meios acadêmicos através de diversas iniciativas.

O foco destas iniciativas é oferecer grandes capacidades de processamento e de armazenamento, utilizando técnicas de processamento paralelo (COULOURIS, 2008).

2.2 ALTA DISPONIBILIDADE

Segundo Coulouris (2008), o *cluster* de alta disponibilidade é normalmente

construído com a intenção de fornecer um ambiente seguro contra falhas utilizando-se da redundância de componentes (sejam de *hardware*, *software*, serviços de rede ou de interconectividade ou interoperabilidade). Em outras palavras, fornecer um ambiente computacional onde a falha de um ou mais componentes não irá afetar significativamente a disponibilidade do ambiente de computação ou aplicações que estejam sendo usadas.

2.3 BALANCEAMENTO DE CARGA

O *cluster* de balanceamento de carga é usado para fornecer uma interface simplificada para um conjunto de recursos que podem aumentar ou diminuir com o passar do tempo e conforme a necessidade por processamento. Neste tipo de *cluster*, estão implícitos os conceitos da alta disponibilidade (com a redundância de componentes) e de alto desempenho de computação (com a distribuição das tarefas completas pelos vários componentes replicados ao nó central, montando uma espécie de quebra cabeça (CA-SAVANT, 1988; SINGHAL, 1992).

2.4 ETHERNET E TCP/IP

Como cita Perry (2005), *ethernet* vem sendo a principal tecnologia de redes locais desde o início dos anos oitenta, e é usada por todas as empresas, que tenham uma infraestrutura mínima de rede. Embora os seus protocolos básicos tem mudado pouco, novas opções como o *Fast Ethernet* (100Mbps) e o *Gigabit Ethernet* (1 Gbps), aliados aos baixos preços dos equipamentos tem garantido uma sobrevida fenomenal ao *Ethernet*.

Segundo Perry (2005), o nome oficial do padrão *Ethernet* é *IEEE 802.3*, padrão este que foi publicado em 1985 com o título: *IEEE 802.3*, já o suplemento 802.3i-1990 descreve a utilização do 802.3 em banda base a 10Mbps com par trançado categoria 3 ou superior (10Base-T), o qual é implementado pelo controlador 802.3 usado.

Sistemas 10 Base-T como o nosso funcionam muito bem na grande maioria das redes, mas é recomendada a utilização de switches ou switching-hub, para isolar a rede 10Base-T das outras de maior velocidade (MIERS, 2003).

O *TCP/IP* é uma suíte de protocolos que permite que computadores de todos os tipos, rodando sistemas operacionais totalmente distintos comuniquem entre si. Surgiu como uma pesquisa financiada pelo governo de redes por chaveamento de pacotes e acabou se tornando a maior estrutura tecnológica já criada pela humanidade, envolvendo o mundo inteiro. É um sistema aberto no sentido de que seus padrões são divulgados e podem ser implementados livres de qualquer ônus. Estas facilidades levaram ao surgimento da Internet (PERRY 2005).

O *TCP/IP* é uma suíte de protocolos de quatro camadas apenas: enlace, rede, transporte e aplicação. O controlador ethernet; seu driver e o protocolo *ARP* formam a camada de enlace no nosso caso (PERRY, 2005).

A ampla aceitação e utilização da *Ethernet* e do *TCP/IP*, levaram algumas empresas a implementar estes protocolos em chips de baixo custo para aplicações embutidas, devido ao surgimento de tais chips a aplicação se tornou viável (MIERS, 2003).

2.5 FERRAMENTAS DE RENDERIZAÇÃO DE IMAGENS

Sabe-se que a origem da palavra renderizar é originária da palavra inglesa render sendo um termo bastante usado na computação gráfica para fazer menção ao convertimento de um tipo de arquivo para outro, ou até mesmo a "tradução" de uma linguagem para outra (GONZALEZ, 1993).

Nesse processo de renderização de uma determinada cena é preciso fazer definição de muitas "variáveis", por exemplo os objetos presentes na cena, suas texturas, sua cor, reflexão, transparência e fazer a localização de pontos de iluminação. Pode-se resumir o processo de renderização como sendo uma forma de converter modelos gráficos em uma imagem. Nesse ato de renderizar, os programas responsáveis, fazem cálculos diversos, como da perspectiva dos planos, das sombras e luzes do objeto (PLOE, 2006).

O principal motivo da utilização de renderização nessa monografia é o fato desse ato sempre exigir uma grande capacidade computacional, podendo tirar proveito, até certo ponto, dos dois temas focos: *clusters* e arquiteturas multicore. Esse capítulo não tem por objetivo se aprofundar nos tipos de ferramentas de renderização, pois esse não é o foco do trabalho, e sim fazer uso dessas ferramentas para servirem de estudo para *clusters* e arquiteturas multicore (GONZALEZ, 1993).

3 METODOLOGIA

Neste capítulo é abordado a parte prática da monografia, com aspectos relacionados às atividades desenvolvidas e os resultados obtidos. Assim, serão descritas as configurações do *cluster*, e a parte dos testes que envolvem as ferramentas de renderização, as várias estruturas e consequentemente os resultados e conclusões obtidas após os testes realizados.



Figura 8 – Estrutura pronta do Cluster.

Fonte: Elaborado pelo autor.

A Figura 8, mostra a estrutura física do *cluster*, composta por dois computadores de mesa, um *notebook*, um chaveador, um roteador e um monitor padrão CRT.

3.1 ESTRUTURA DO CLUSTER

Foi elaborado uma estrutura de *clusters*, que foi comparado a uma máquina normal, com a finalidade de realizar testes com a ferramenta de renderização. O número máximo de nós utilizados foram 3, onde o *notebook* era o nó mestre gerenciando as entradas de processamento e mais dois PC's de mesa como escravo, com as seguintes configurações:

<u>Nó Master, um Notebook:</u>	
CPU	Intel i3 core CPU
Memoria Cachê	4MB
Memoria Ram	4 GB DDR2 800 MHz
Placas de rede	10/100/1000
Disco rígido	500 GB SATA

Figura 9 – Nó Mestre.
Fonte: Elaborado pelo autor.

Na Figura 9 descreve em detalhes a parte de *hardware* do nó mestre do cluster.

<u>1º nó escravo, um PC de mesa:</u>	
CPU	Intel Cor 2 Duo 2,33 GHz
Memoria Cachê	1 MB
Memoria Ram	3 GB DDR 667 MHz
Placas de rede	10/100/1000
Disco rígido	1,5 Tera Sata

Figura 10 – 1º Nó Escravo.
Fonte: Elaborado pelo autor.

Na Figura 10 descreve em detalhes a parte de *hardware* do 1º nó escravo do cluster.

<u>2º nó escravo, um PC de mesa</u>	
CPU	AMD 2,1 GHz
Memoria Cachê	512 MB
Memoria Ram	1 GB DDR 667 MHz
Placas de rede	10/100/1000
Disco rígido	500GB 7200RPM Sata

Figura 11 – 2º Nó Escravo.
Fonte: Elaborado pelo autor.

Na figura 11 descreve em detalhes a parte de hardware do 2º nó escravo do cluster.

De forma a gerenciar todos os nós, obteve-se um chaveamento KVM 4 portas, modelo TK 400, um dispositivo eletrônico que auxilia no comando de uma só vez 4 computadores simultaneamente, com objetivo de diminuir a quantidade de monitor, mouse e teclado e consequentemente, ter um maior espaço físico.



Figura 12 – Chaveador.
Fonte: Elaborado pelo autor.

Como mostra na Figura 12, um chaveador, onde através desta tecnologia foi feito o controle de dois computador de mesa e um notebook, com apenas um monitor CRT.

Os nós pertencentes a todas as estruturas do cluster eram homogêneos com as configurações descritas na Figura 13.

INFRAESTRUTURA DE REDE	
Fast Ethernet	10/100 Mbps
Switch	10/100 8 portas
Cabo par trançado	10/100
Conector	RJ45
Protocolo	TCP/IP

Figura 13 – Infraestrutura de Rede.
Fonte: Elaborado pelo autor.

Nós com configurações iguais não são necessários, mas já ajudam a evitar, por exemplo, problemas decorrentes de compilação de um código para arquiteturas diferentes, ou por ter também uma carga de trabalho desbalanceada, onde os nós mais rápidos irão terminar primeiro que os mais lentos, de modo que os mais rápidos terão que ficar esperando. Assim, a homogeneidade não é uma condição necessária para um *cluster*, mas é um fator que irá reduzir bastante à quantidade de problemas.

Os nós estão interligados por um *switch* 10/100 Mbps e por cabos padrão *Ethernet*.

Logo abaixo estaremos visualizando os projetos renderizado no *cluster*, a fim de quantificar sua capacidade de processamento.

Na Figura 15 temos o 1º exemplo de renderização, onde temos uma quarto em 3D com algumas caixas, tendo 900 *frames*, no formato *blender*.

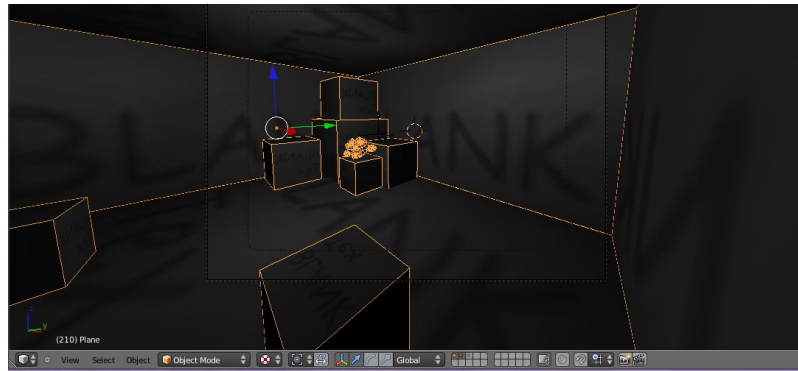


Figura 15 – Exemplo 1 quarto geométrico (0.413GB).

Fonte:Blender ([1995]).

Nota: Adaptado pelo autor.

Já na Figura 16 tem-se o 2º exemplo de renderização, onde uma sala mais elaborada com várias texturas, algumas caixas e uma arma, simula um ambiente de jogo, tendo assim 1000 *frames*, no formato *blender*.



Figura: 16 – Exemplo 2 galpão de guerra (0.850GB).

Fonte:Blender ([1995]).

Nota: Adaptado pelo autor.

A Figura 17 com o 3º exemplo de renderização, tem-se um uma sala com uma escadaria toda em texturas, parede com texturas e iluminação, a frente uma esfera, tendo 1.148 *frames*, no formato *blender*.

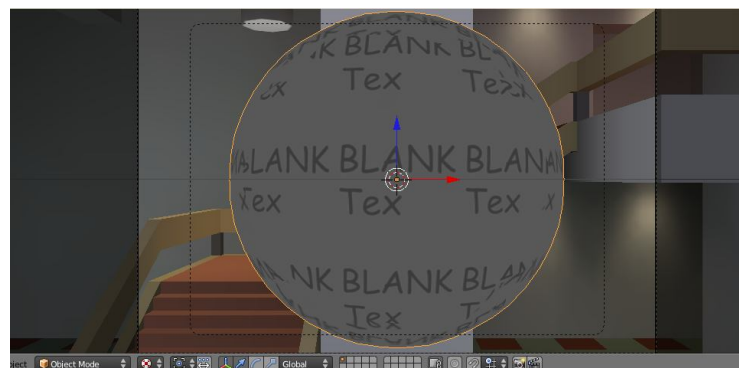


Figura: 17 – Exemplos 3 sala cenográfica (1.900GB).
 Fonte:Blender ([1995]).
 Nota: Adaptado pelo autor.

Na Figura 18 tem-se o 4º exemplo para renderização, com um chão de textura e efeitos, um boneco com várias formas geométricas, cores e textura, tendo assim 1.350 *frames*, no formato *blender*.

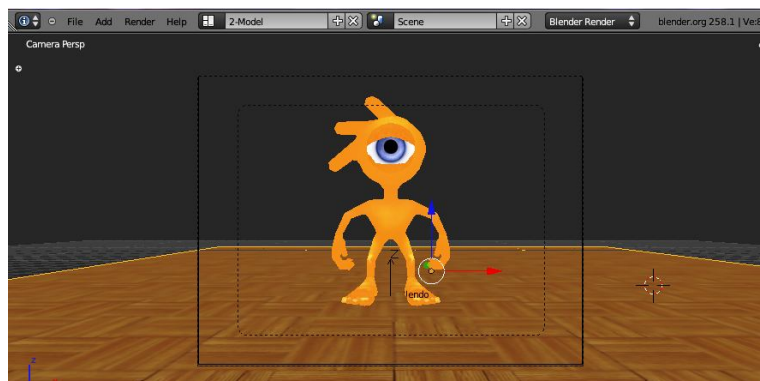


Figura: 18 – Exemplo 4 boneco blender (2.010GB).
 Fonte:Blender ([1995]).
 Nota: Adaptado pelo autor.

3.2 CAPACIDADE DE RENDERIZAÇÃO DO CLUSTER

Através da utilização da ferramenta e coleta de dados, das figuras exemplo 1 quarto geométrico, exemplo 2 galpão de guerra, exemplo 3 sala cenográfica e exemplo 4 boneco blender, citado nas paginas 39 e 40, foi identificado o poder computacional do *cluster*, como mostra a Tabela 2.

Tabela: 2 – Tabela de calculo computacional.

Arquivos	Frame	Taman. dos Proj. GB	Processos
Exemplo 1	900	0,413	13.148
Exemplo 2	1.000	0,850	13.594
Exemplo 3	1.148	1,900	15.356
Exemplo 4	1.350	2,010	18.092
Total	4.398	5,173	60.190
Média	1.100	1,293	15.048

Fonte: Elaborado pelo autor.

Na Tabela 2 é composta pelas seguintes colunas, arquivos, a descrição dos arquivos que foram utilizados, já na coluna *frame* tem-se a quantidade de *Frames* de cada exemplo, e na 3ª e última coluna processo, extraídos do monitoramento do *DrQueue*, onde mostra a quantidade de processo executado para cada exemplo apresentado.

Baseado na quantidade de 1.100 Frame como mostra na tabela 2, 3ª coluna, 6ª linha, como desvio padrão, a fim de garantir o cálculo; chegou-se a uma capacidade computacional do cluster de 15.048 processos como mostra na tabela 2, 3ª coluna, 6ª linha, como desvio padrão, para cada 1.100 frames apresentados.

De forma a mensurar o quanto de tempo que se ganharia com a construção do cluster para renderização de imagem.

Lembrando que este teste não foi o foco do estudo, mas sim uma forma de mensurar o quanto significava o ganho na renderização com cluster em uma máquina comum.

Nos testes realizados, foram feitas variações na quantidade de nós, conforme mostra as tabelas 3 e 4. Na tabela 3, na 3ª coluna com o cluster completo com 3 nós renderizando. Os primeiros testes foram feitos com quatro exemplos, conforme mostra a tabela 3 na 2ª coluna.

Tabela: 3 – Numeros do 1º Teste com Cluster.

Arquivos	Frame	Taman. dos Proj. GB	Nós
Exemplo 1	900	0,413	3
Exemplo 2	1.000	0,850	3
Exemplo 3	1.148	1,900	3
Exemplo 4	1.350	2,010	3
Total	4.398	5,173	12
Média	1.100	1,293	3

Fonte: Elaborado pelo autor

No segundo teste foram utilizados os mesmos quatro exemplos como mostra a tabela 4, na 1ª coluna, porém com um só nó renderizando, conforme mostra na tabela 4, 3ª coluna.

Tabela: 4 – Numeros do 2º Teste sem Cluster.

Arquivos	Frame	Taman. dos Proj. GB	Nós
Exemplo 1	900	0,413	1
Exemplo 2	1.000	0,850	1
Exemplo 3	1.148	1,900	1
Exemplo 4	1.350	2,010	1
Total	4.398	5,173	4
Média	1.100	1,293	1

Fonte: Elaborado pelo autor

Como mostra na tabela 5, foram feitas 8 séries de testes e tirado o desvio padrão entre eles, para aumentar a confiabilidade nos resultados apresentados. O tempo total de simulação no *cluster* foi de aproximadamente de 17 horas de processamento, como mostra na tabela *com cluster* 4ª coluna 6ª linha, já na máquina comum mais de 66 horas de processamento, como mostra na tabela *sem clusters* 4ª coluna 5ª linha.

Tabela 5 – Tempo de Processamento em Horas do Cluster Versos Sem Cluster

Arquivos	Frame	Processos	Taman. dos Proj. GB	Tempo Horas	Nós
Exemplo 1	900	13.148	0,413	3,7	3
Exemplo 2	1.000	13.594	0,850	3,8	3
Exemplo 3	1.148	15.356	1,900	4,3	3
Exemplo 4	1.350	18.092	2,010	5,0	3
Total	4.398	60.190	5,173	16,7	12
Média	1.100	15.048	1,293	4	3

Sem Cluster

Arquivos	Frame	Processos	Taman. dos Proj. GB	Tempo Horas	Nós
Exemplo 1	900	3.984	0,413	14,6	1
Exemplo 2	1.000	4.119	0,850	15,1	1
Exemplo 3	1.148	4.653	1,900	17,1	1
Exemplo 4	1.350	5.482	2,010	20,1	1
Total	4.398	18.239	5,173	66,9	4
Média	1.100	4.560	1,293	17	1

Fonte: Elaborado pelo autor

ESTUDO DE CASO 1:

Numa primeira situação foi adotada como critério a divisão de processos iguais ao número de máquinas, sendo assim, uma máquina, um processo, duas máquinas, dois processos, assim por diante.

Tabela 6 – Resultados obtidos no processamento dos dados.

Arquivos	Frame	Taman. dos Proj. GB	Processos
Exemplo 1	900	0,413	13.148
Exemplo 2	1.000	0,850	13.594
Exemplo 3	1.148	1,900	15.356
Exemplo 4	1.350	2,010	18.092
Total	4.398	5,173	60.190
Média	1.100	1,293	15.048

Fonte: Elaborado pelo autor.

Através dos dados apresentados acima é possível notar que ocorreu um aumento gradativo nas quantidades de processamentos, conforme mostra a 3ª coluna, coletado do monitoramento do extraídos do monitoramento do *DrQueue*, fazendo com que se exigisse cada vez mais do processamento do *cluster*.

Um segundo teste foi analisado dentre os processos, dos quatros exemplos a observar em quanto tempo e quanto que seria consumido de *cpu* e *memória ram* de todo o *cluster*, conforme apresentado na tabela 7.

Tabela 7 – Resultados obtidos na eficiência do Cluster.

Arquivos	Tempo Horas	CPU	Memoria Cons. MB	Nucleo	Memoria Total MB	Nós
Exemplo 1	3,7	66%	4.383	6	8.192	3
Exemplo 2	3,8	68%	4.531	6	8.192	3
Exemplo 3	4,3	77%	5.119	6	8.192	3
Exemplo 4	5,0	90%	6.031	6	8.192	3
Total	16,7	301%	20.063	24	32.768	12
Média	4	75%	5.016	6	8.192	3

Fonte: Elaborado pelo autor.

Com este levantamento, como pode ser observado na tabela 7, obteve-se um tempo aproximadamente de 16 horas de renderização, conforme mostra a 2º coluna, 5º linha, um desvio padrão de *cpu* de 75% de uso, conforme mostra a 3º coluna, na 6º linha, e por fim um consumo de *memória ram*, em desvio padrão de mais de 5 mil *MB*, como mostra na 4º coluna, 6º linha, já na 5º coluna obteve-se a quantidade de núcleos sendo utilizados, na 6º coluna o total de *memória ram* disponível no *cluster* e 7º coluna a quantidade de nós sendo executadas, em todas estas quatro renderizações.

Nos estudos de caso com o *cluster* foram feitas variações manuais dentro do extraídos do monitoramento do *DrQueue* na quantidade de processos com a finalidade de se investigar alguma vantagem em fazer o aumento do número de processos, de forma a explorar ao máximo todos nós do *cluster*, a fim de chegar em mais de 60 mil processos, de forma a tentar baixar as 16 horas de processamento, no entanto mesmo com os *mixer* de configurações não conseguimos obter uma melhora significativa, nos processamentos e tempos de processamento.

ESTUDO DE CASO 2

No estudo de caso 2 descreve os dados detalhados onde obteve-se com a renderização feita a partir de uma máquina com as seguintes configurações:

CPU – AMD 2,1 GHz

Memória cachê – 512 MB

Memória Ram – 1 BG 667 MHz

Disco rígido – 500 GB 7200 RPM Sata

O qual obteve-se os seguintes resultados de processamento de dados, conforme listado na tabela 8.

Tabela 8 Resultados obtidos no processamento dos dados.

ComCluster

Arquivos	Frame	Processos	Taman. dos Proj. GB
Exemplo 1	900	13.148	0,413
Exemplo 2	1.000	13.594	0,850
Exemplo 3	1.148	15.356	1,900
Exemplo 4	1.350	18.092	2,010
Total	4.398	60.190	5,173
Média	1.100	15.048	1,293

Sem Cluster

Arquivos	Frame	Processos	Taman. dos Proj. GB
Exemplo 1	900	3.984	0,413
Exemplo 2	1.000	4.119	0,850
Exemplo 3	1.148	4.653	1,900
Exemplo 4	1.350	5.482	2,010
Total	4.398	18.239	5,173
Média	1.100	4.560	1,293

Fonte: Elaborado pelo autor.

Na tabela 8 na coluna arquivos tem-se a descrição de cada exemplo, já na coluna frame tem-se a quantidade de *frames* de cada exemplo, na coluna processo tem-se a quantidade de processamento executados para cada exemplos apresentados.

Como é possível verificar, uma queda de mais de um terço no processamento do desvio padrão, conforme mostra a 3º coluna, 6º linha da tabela sem *cluster*, comparada com a 3º coluna, 6º linha da tabela com *cluster*.

Em uma outra análise dentre os processos, dos quatros exemplos a observar em quanto tempo e quanto seria consumido de cpu e memoria ram, da máquina comum, conforme apresentado na tabela 9.

Tabela: 9 Resultados obtidos na eficiência da maquina comum.

Arquivos	Tempo Horas	CPU	Memoria Cons. MB	Nucleo	Memoria Total MB	Nós
Exemplo 1	14,6	99%	1.003	1	1.024	1
Exemplo 2	15,1	99%	1.009	1	1.024	1
Exemplo 3	17,1	99%	1.010	1	1.024	1
Exemplo 4	20,1	99%	1.020	1	1.024	1
Total	66,9	396%	4.042	4	4.096	4
Média	17	99%	1.011	1	1.024	1

Fonte: Elaborado pelo autor

Como pôde-se analisar nos resultados obtidos na tabela 9, um grande consumo de *cpu* de 99% como mostrado na 3º coluna, 6º linha, como desvio padrão, e um outro dado relevante é com relação ao consumo de *memória ram*, onde obteve-se um consumo como desvio padrão de 1.011 MB, como mostra na, 4º coluna, 6º linha, onde se leva em conta que tinha-se um total de 1.024 MB, como mostra na 6º coluna, 6º linha.

Vale apenas ressaltar que em vários momentos deste teste observou-se alguns travamentos por parte do alto consumo de processamento sendo executados, tomando como consequência, mais de 67 horas para processar os mesmos quatro exemplos utilizados no *cluster*, como mostra na, 2ª coluna, 5ª linha.

4 CONSIDERAÇÕES FINAIS

Neste trabalho foi possível concretizar um estudo do ambiente de computação em *cluster* utilizando como sistema operacional *Ubuntu Server 11 x86*, *DrQueue* e o *Blender*, com a montagem de uma estrutura e configuração de um *cluster* utilizando esta tecnologia. Executando aplicações de renderização de imagem no *cluster*, pôde-se constatar que aplicações como estas que exigem grande poder de processamento podem tirar vantagens de uma estrutura de *cluster*, baseado em computadores pessoais.

Um dos pontos fundamentais que foram observados com o desenvolvimento deste trabalho, é a diferença notória de uma máquina comum, para um *cluster* com as mesmas máquinas comuns, porém, trabalhando de forma paralelas em rede. Um outro ponto bastante notório é o grande trabalho de paralelismo do *software DrQueue*, ele não paraleliza a aplicação que será executada, identifica um nó mais livre na rede, e migra do contexto de *software* para outro nó, balanceando as aplicações de forma a ter como requisitos a compatibilidade com a migração de processos, para tirar proveito do *cluster*.

No capítulo de estudo da ferramenta de *renderização* chamada *Blender*, nota-se que este *software* de criação e renderização de imagem, conseguiu tirar proveito da estrutura de *cluster* montada, dentre todas as animações *renderizadas*, onde obteve grande resultado na sua distribuição dentro dos nós.

Neste teste, o foco foi visualizar o quanto se obtinha de processamento num *cluster* com três máquinas *renderizando*, partindo deste propósito, colhe-se os resultados, de forma a saber o quanto o *cluster* renderia para cada *frame* colocado para renderizar, onde obteve-se grandes resultados com um desvio padrão de cada 1.100 *frames* compilados, tinha-se um desvio padrão de 15.048 de processamento renderizado, com um desvio padrão de tempo de 4 horas de processamento, ou seja, tendo a cada hora *renderizada* 275 *frames*, sendo assim a cada hora *renderizando* 25% de todo o seu trabalho.

Além do estudo e testes realizados, foi executado uma comparação de quanto tempo de renderização um *cluster* efetivamente ganhou, ao invés de fazer em uma máquina comum. Obtendo os seguintes números, para os mesmos arquivos processados no *cluster* colocado em uma máquina comum, obteve-se uma perda de 50,16 horas a mais, enquanto todos estes arquivos no *cluster* *renderizava* em 16,7 horas em uma máquina comum, os mesmos arquivos demoravam para *renderizar* 66,88 horas.

Como trabalho futuro pode-se pensar em uma maior quantidade de máquinas, de modo a ganhar maior poder computacional, em paralelo buscar um melhor tráfego de rede, de forma a expandir os 10/100 Mbps de tráfego, obtido neste projeto, visto que dentro do *cluster* como cita Pitanga (2004), um dos maiores problemas de performance é a rede, devido ao grande número de *delay* (colisões de pacotes).

REFERÊNCIAS

ALECRIM, E. Cluster: principais conceitos. Info Wester, 2004. Disponível em; <
<http://www.infowester.com/cluster.php>>. Acesso em: 24 jun. 2011.

BUYYA, Rajkumar: High. Performance Cluster Computing: Architectures and Systems. Volume 1. New Jersey : Prentice-Hall, 1999. 849p.

Gonzalez (1993) Gonzalez, R. F.; Woods, R. E. Digital Image Processing. Addison-Wesley, p 716. 1993.

MACHADO, M., Hofmam, M. Sistemas distribuídos MPI, ed. 3ª Editora. São Paulo 2005.

PITANGA, Marcos. Computação em Cluster: O Estado Arte da Computação. Rio de Janeiro: Brasport, 2003.

PITANGA, Marcos. Construindo supercomputadores com linux. 2ª edição. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2004. 292p

(Perry 2005) Perry S. Marshall e John S. Rinaldi Industrial Ethernet 2nd Edition

[PLOE] Atze van der Ploeg. Interactive Ray Tracing, the replacement of rasterization. Tese B.Sc., Vrije Universiteit, Dezembro 2006.

YOKOKURA, Alex Yuichi, Estudo de Viabilidade da Implantação de Técnicas de Cluster ao Projeto Servidor de Estações de Trabalho 2007 (SET).