

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/251127307>

REAPROVEITAMENTO DE COMPUTADORES ATRAVÉS DE MÁQUINAS VIRTUAIS PARALELAS (PVM)

Article · June 2012

CITATIONS

0

READS

410

1 author:



[Ezequiel Ferreira Santos](#)

Sabesp

24 PUBLICATIONS 17 CITATIONS

SEE PROFILE

REAPROVEITAMENTO DE COMPUTADORES ATRAVÉS DE MÁQUINAS VIRTUAIS PARALELAS (PVM)

Luciano Kazuo Tamehiro, Marcio Suekasu Takahashi e Ezequiel Ferreira dos Santos

Universidade Nove de Julho, São Paulo/SP, Brasil

E-mail: luciano.kazuo@gmail.com, suekasu@hotmail.com, ezequiel.santos6@fatec.sp.gov.br

Resumo: O objetivo deste artigo é efetuar uma análise da utilização da computação paralela através da biblioteca para passagem de mensagens PVM (Parallel Virtual Machine) visando o baixo custo, com o uso de microcomputadores obsoletos, sistemas e softwares livres. São escassas as pesquisas que avaliam a utilização de computação paralela na reciclagem de microcomputadores, principalmente no Brasil. A pesquisa foi baseada em análise bibliográfica, levantamento histórico e de uso de microcomputadores no Brasil, diferentes métodos de reciclagem e testes entre três microcomputadores. Os resultados mostraram que a utilização do PVM aperfeiçoa o tempo de execução de determinada tarefa e pode ser uma alternativa para a reciclagem de microcomputadores obsoletos, possibilitando a utilização de vários microcomputadores como um supercomputador para aplicações específicas, como a criação de imagens em 3D na computação gráfica.

Palavras Chaves: Ampliação do tempo de vida útil, sistemas computacionais paralelos.

I. Introdução

Com o desenvolvimento de processadores mais eficientes e aplicações cada vez mais robustas, que necessitam de maior desempenho, resultou no descarte cada vez mais rápido de microcomputadores obsoletos, gerando um grande problema ambiental: o crescimento de resíduos tecnológicos.

O propósito deste trabalho é analisar uma melhor destinação desses microcomputadores através do conceito de computação paralela utilizando a biblioteca para passagem de mensagens PVM (Parallel Virtual Machine), aumentando o desempenho e o processamento, já que ela permite que um microcomputador central redistribua as tarefas para outros microcomputadores simulando um único computador mais robusto.

Como decorrência do propósito geral, os objetivos específicos a serem explorados são: histórico do desenvolvimento do hardware, identificar as tecnologias que estão à disposição para a reciclagem ou reutilização e descarte desses equipamentos; pesquisar as principais práticas para ampliação da vida útil [1].

II. Consumo e descarte de microcomputadores no Brasil

Os microcomputadores já fazem parte do dia-a-dia de grande número de usuários no Brasil. Com o câmbio favorável e a tendência de redução dos preços, o consumo de novas tecnologias ampliou significativamente, como mostra a Tabela I. Houve um aumento de 10 milhões de equipamentos de informática nos três primeiros anos, entre 2004 a 2007. O mesmo ocorreu em um intervalo menor entre 2007 a 2009, ou seja, a produção de equipamentos tende a ser maior em um intervalo de tempo menor.

Tabela I - Produção anual de equipamentos de informática no Brasil da empresa IT Data consultoria, 2009.

Segmento	2004	2005	2006	2007	2008	2009
Doméstico	5.234.347	6.406.036	8.662.000	11.710.000	15.150.000	18.200.000
Corporativo	11.143.630	11.900.529	13.614.000	15.450.000	17.310.000	19.190.000
Servidores	274.850	298.210	325.384	363.090	392.500	414.000
Total	16.652.827	18.604.775	22.601.384	27.523.090	32.852.500	37.804.000

Assim como o consumo disseminado de computadores é recente, o surgimento de resíduos tecnológicos é também um fato novo[2]. Desde o início da década, cerca de 80 milhões de microcomputadores se tornaram obsoletos causando problemas ao meio ambiente em decorrência de sua destinação incorreta.

A atualização ou substituição dos equipamentos obsoletos não são analisadas por ferramentas ou metodologias que possibilitam o reuso deles em outras aplicações [3]. A necessidade de processamento mais ágil, maior capacidade de armazenamento e crescimento na demanda de transferência de dados que as novas aplicações exigem é o principal fator no descarte dos computadores no Brasil.

A atualização de um microcomputador deve ser feita após verificar a possibilidade da troca de memória de acesso randômica (memória RAM), disco rígido e processadores para melhoria da performance. Quando possível, a troca de peças permite uma sobrevida do computador, melhorando o seu desempenho. Uma simples atualização pode atender uma nova aplicação que antes não era possível usar nesta máquina.

Quando a atualização não é satisfatória, existem alternativas para reaproveitamento deste material.

Um exemplo de reutilização de computadores no Brasil é o projeto da Universidade de São Paulo (USP) do Centro de Computação Eletrônica (CCE) chamado

Centro de Descarte e Reúso de Resíduos de Informática (CEDIR) que recebe peças e equipamentos de informática e eletrônicos obsoletos. O equipamento é analisado e as peças que podem ser reaproveitadas são usadas para montar microcomputadores que são utilizados em programas de inclusão digital em projetos sociais. As peças que não tem utilidade são enviadas para empresas especializadas em reciclagem.

Materiais como placa-mãe e memória de acesso randômico (memória RAM) podem passar por técnicas de reaproveitamento do material composto. Pode-se utilizar a técnica de lavagens e tratamentos térmicos (química de reconsumo), que através de reações químicas em forno calcinador de altas temperaturas efetua a retirada de materiais, como o ouro e o cobre.

Outra técnica utilizada é a moagem (moer, dividir em pedaços extremamente pequenos), que consiste na transformação do material em pó [4]. Após o processo, este pó é separado por material (ouro, cobre, fibra de vidro), onde serão reaproveitados para outros fins.

Os primeiros computadores eram máquinas grandes e caras, como o *Electronic Numerical Integrator and Computer* (ENIAC), apresentado em 1946, que ocupava 167m² e pesava em torno de 27 toneladas e criada por John Presper Eckert e John Willian Mauchly, e realizava 5000 somas/segundo tinha 18000 válvulas.

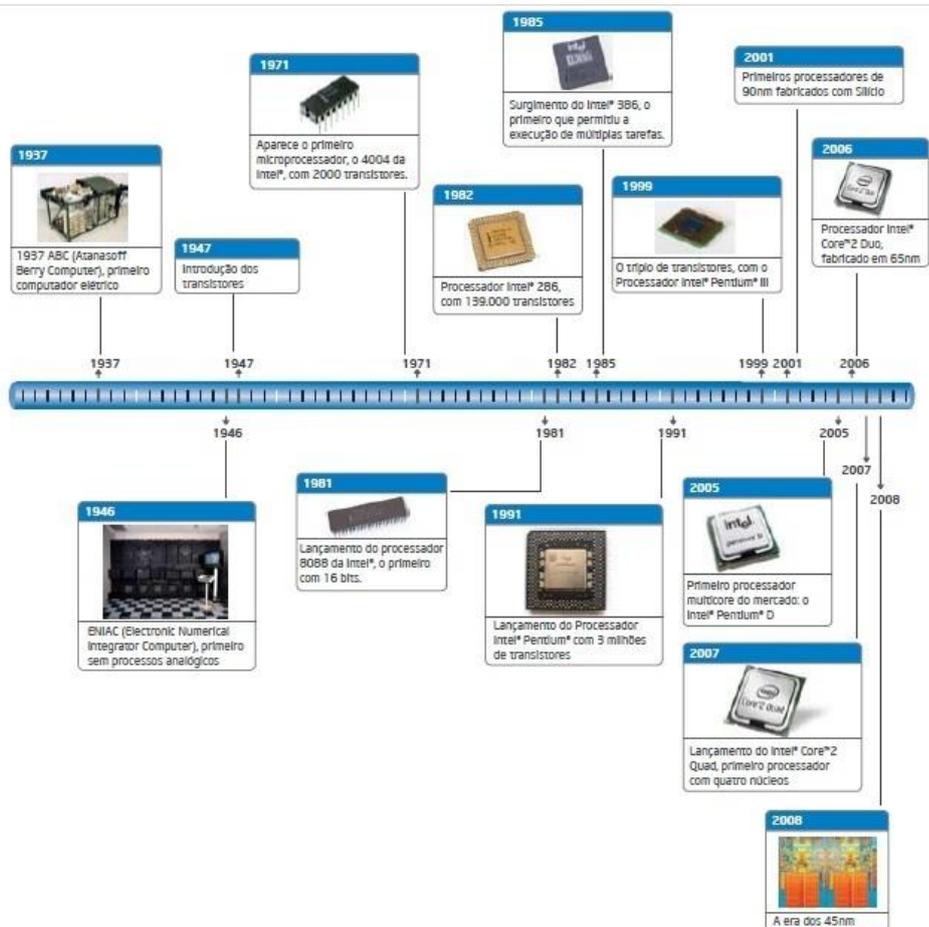


Figura I – Linha de evolução dos processadores Intel [6].

Na década de 50 a válvula foi substituída pelo transistor, inventado por Willian Shockley, John Bardeen e Walter Brattain, que são pequenos interruptores de transmissão elétrica e possuem uma maior durabilidade, menor consumo e com dimensões reduzidas em relação as válvulas, permitindo o surgimento dos primeiros computadores comerciais, como IBM 1401 com 4096 *bytes* de memória e processava 193.300 somas por segundo (decimais de 8 dígitos) e custava o equivalente a U\$2500/mês na configuração mínima.

Na décadas de 60 e 70 com a introdução dos circuitos integrados, permitiu o aparecimento do primeiro microprocessador, o 4004 da Intel que continha 2000 transistores e a venda de um dos primeiros computadores pessoais (ou o primeiro a ser entregue montado), 50 unidades do Apple I que foram vendidas por U\$ 666,66 em abril de 1976 com um processador de 1 MHz [5]. A Figura I ilustra a evolução dos processadores da Intel.

Em 1981 o processador Intel 8088, já dispunha de 10 MHz, 29.000 transistores e já possuía a arquitetura de 16 *bits*, um anos depois o processador Intel 286 possuía 39.000 transistores e 1985 foi lançado o Intel 386 com arquitetura de 32 *bits*, permitindo a execução de tarefas múltiplas e com o desenvolvimento de interfaces gráficas houve a popularização dos computadores pessoais e a disseminação do modelo cliente/servidor.

Na década de 90, existe um grande avanço dos processadores, em 1991 o processador Intel *Pentium* tinha 3 milhões de transistores e em 1999 o processador Intel *Pentium3* continha 9,5 milhões e foi um dos primeiros processadores fabricado com silício.

Em 2005 surgiu o primeiro processador *multicore* do mercado, o Intel *Pentium D* que foi o começo da tecnologia *Dual-Core*, que levou ao desenvolvimento do processador Intel *Core 2 Duo* fabricado com tecnologia de 65nm e com até 290 milhões de transistores.

A tecnologia de 45 nm possibilitou o surgimento dos menores processadores do mundo, como Intel *Core 2 Duo*, Intel *Core 2 Quad*, Intel *Xeon* e também a linha mais recente de processadores o Intel *Atom* voltado para o economia de energia e otimização do uso do processador.

III. Computação paralela

A computação paralela teve início na década de 70 com a introdução dos sistemas distribuídos que são sistemas nos quais os componentes de hardware e software localizados em uma rede de computadores se comunicam e coordenam suas ações somente por troca de mensagens [7].

Tem como principal função dividir a tarefa principal em subtarefas que são executadas simultaneamente em processadores diferentes. Essas subtarefas são

distribuídas inteiramente, sem modificação do código. [3].

A comunicação pode ser feita por troca de mensagens de um processo para o outro ou memória compartilhada, onde os dados ficam armazenados em *buffers*¹ temporários, permitindo a portabilidade.

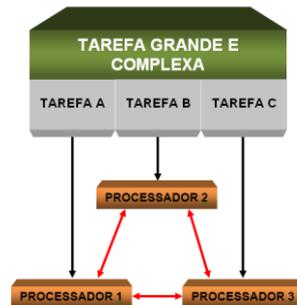


Figura II – Modelo de computação paralela

Um sistema de computação paralela possui tarefas (ou processos), que são programas executados, concorrentemente, geralmente disparados por um programa mestre [1]. As tarefas são as principais unidades do processamento paralelo em um ambiente de computação distribuída. Comunicam-se através de troca de mensagens.

Um programa sequencial sofre transformação para um programa paralelo (paralelização), com a identificação de porções de código que podem ser executados independentemente. Exige mudanças no código do programa e, caso necessário, no algoritmo utilizado no programa sequencial.

A paralelização consiste na separação dos dados para serem processados em paralelo, sendo que após o processamento, são reunidos novamente, otimizando os processadores múltiplos (paralelismo de dados) e possui características como o uso da aceleração (*speed-up*) que consiste na comparação entre o tempo de execução do programa em um único processador e o tempo de execução utilizando vários processadores e da sincronização que é a coordenação necessária entre processos para a troca de informações, podendo ser um fator de decréscimo da eficiência do programa, uma vez que alguns processadores podem ficar inativos, esperando pelo término de outros processos.

O sistema é considerado escalável se a aceleração atingida cresce proporcionalmente ao número de processadores utilizados.

A granularidade é determinada quando uma quantidade de processamento realizado por cada processo, em relação à quantidade de comunicação entre processos. Quando os processos executam poucas instruções e necessitam se comunicar muito, diz-se que o programa é muito granular. Quando, ao contrário, os processos executam muitas instruções, com pouca troca de informação, diz-se que o programa é pouco granular.

¹ Buffer: Região temporária da memória de acesso rápido para leitura e escrita de dados.

Um programa com granularidade alta necessita de uma maior sincronização que um programa com menor granularidade, o que afeta o tempo de execução do programa.

O balanceamento de carga consiste na distribuição equilibrada de tarefas entre os processadores, de forma a garantir uma execução eficiente do programa paralelo.

De maneira geral, os modos de programação paralela podem ser classificados em três modelos [8]: o paralelismo de dados, que é uma técnica que separa os dados para serem processados em paralelo, sendo que posteriormente são reunidos novamente, otimizando os processadores múltiplos; o paralelismo de tarefas ocorre quando as tarefas são independentes umas das outras podendo ser executadas por processadores diferentes. É a técnica mais simples de programação paralela e por último, o paralelismo de objetos que usa o conceito de objetos sendo acessados por métodos em diferentes computadores para fins específicos.

A computação paralela auxilia na capacidade de solucionar alguns entraves atuais de desempenho e tecnologias, como problemas complexos que exigem alto poder computacional, já que permite o uso de vários computadores para gerar um supercomputador e o aumento da velocidade dos processadores que esbarra no alto consumo de energia, geração de calor e no desenvolvimento de tecnologias com a finalidade de reduzir o tamanho dos componentes, o uso de dois processadores em paralelo é mais barato que dobrar a velocidade.

Os programas paralelos têm como foco a sincronização e comunicação entre processos e a distribuição de trabalho entre os processadores, tornando mais difíceis de desenvolver que os programas sequenciais.

IV. PVM

O *Parallel Virtual Machine* (PVM) é um conjunto integrado de ferramentas de softwares e bibliotecas que emula uma máquina paralela utilizando computadores interconectados de arquiteturas distintas [9].

Esse conjunto é livre e possui os códigos abertos para estudos.

Em 1989 no *Oak Ridge National Laboratory*, foi construído o protótipo PVM 1.0 por *Vaidy Sunermam* e *Al Geist*, para uso interno.

A versão 2.0 foi desenvolvida na Universidade de Tennessee em março de 1991, que chegou até a versão 2.4, após testes e resultados críticos.

A versão 3.3 foi concluída em fevereiro de 1993, sendo que a última atualização está na versão 3.4.6. Essa versão possui suporte para os sistemas operacionais Windows, Unix, Sun e para o uso de clusters Beowulf. Inclui comunicação de contexto, manipuladores de mensagem, mensagens persistentes e interoperabilidade entre clusters Windows NT e Unix.

O PVM obtém o paralelismo externo adicionando uma camada de *software* sobre os protocolos padrão de

comunicação da rede para gerenciar o envio de processos para suas máquinas. O PVM é fornecido como um conjunto de funções agrupadas em uma biblioteca para ser "*linkada*" com códigos fonte gerados na linguagem C/C++ ou FORTRAN [10].

Utiliza um conjunto de computadores heterogêneos como se fosse um único supercomputador. Para isso, é necessário que todas as aplicações dos computadores se comuniquem com a biblioteca criada pelo PVM, que inspeciona a execução de todos os processos e tarefas que contenham a aplicação.

Toda máquina com PVM roda um *daemon*², chamado *pvmd* no sistema. Todo processo que uma máquina A envia para a máquina B tem que passar pelo *pvmd* da máquina A e recebida pela *pvmd* da máquina B que destinará o processo para o destino correto.

As tarefas são identificadas por um número inteiro chamado *task identifier* (TID), que são únicos em toda máquina virtual e são criados pelo *pvmd* local sempre que uma tarefa é gerada. Os TIDs são responsáveis pelo recebimento e envio das mensagens. São divididos em dois tipos de identificadores: *Master* que é responsável pelo disparo das tarefas secundárias (*slave*) via mensagem, transmissão das perguntas e recebimento de respostas. *Slave* é responsável pelo recebimento das perguntas, processamento dos dados, coleta dos resultados e transmissão das respostas

Elas podem ser agrupadas em grupo e numeradas, sendo que cada uma pode entrar e sair dos grupos sem interferir em outras tarefas, nesse processo cada vez que uma tarefa entra em um grupo é atribuído um novo número de identificação no grupo, que começa do zero e vai sendo incrementado a cada elemento.

Se houver falha ou perda de acessibilidade de alguma estação que não seja o mestre, o PVM retira automaticamente do processo [11].

O PVM pode rodar em redes de protocolo *UDP*³ e *TCP*⁴. O protocolo UDP não é orientado a conexão, portanto os dados são enviados sem ter confirmação de recebimento pelo receptor, geralmente é utilizado para envio de pequenas informações. Já o protocolo TCP é orientado a conexão, ou seja, antes de enviar os dados é feito uma comunicação entre o remetente e o destinatário, criando um canal de comunicação.

Assim, o protocolo UDP possui vantagens como escalabilidade e comunicação inicial mais rápida. Já o protocolo TCP tem como vantagem a confiabilidade, já que permite o gerenciamento de problemas de envio e recebimento de dados sem ter que interromper as tarefas do usuário, além de flexibilidade para implementação de métodos de tolerância a falhas.

Outro importante tópico para aperfeiçoar a aplicação é o balanceamento de carga, implantando uma

² Daemon: Programa que roda em segundo plano.

³ UDP: User Datagram Protocol, protocolo simples da camada de transporte, não existe garantia de recebimento.

⁴ TCP: Transmission Control Protocol, protocolo da camada de transporte que garante recebimento dos pacotes.

heurística⁵ que garanta a distribuição homogênea dos processos entre as máquinas que compõem o PVM, buscando o balanceamento entre a eficiência e a adaptabilidade na implementação das bibliotecas.

O PVM pode rodar em servidores interconectados via rede de comunicação, tratando-se de uma arquitetura multicomputador (*cluster*), podendo ser classificadas em: Alta disponibilidade, onde o sistema computacional tem que ficar disponível 24 horas por dia, 7 dias da semana. Alta performance, onde o sistema computacional precisa de alto poder de processamento, para resolver grandes problemas complexos, como o cálculo de trajetória de foguetes.

Cluster é um termo amplamente utilizado que significa um sistema montado com mais de um computador, cujo objetivo é fazer com que todo o processamento da aplicação seja distribuído aos outros computadores, de forma que pareça como se fosse um único computador. Cada computador deve estar interconectado através de uma rede, podendo ser de qualquer topologia [12].

O sistema PVM adquire as características de Single System Image (SSI), onde todos os nós do cluster se tornam uma única máquina virtual. Para que haja a comunicação entre os nós, em cada um haverá um processo chamado *pvm3* que efetua a troca de 14 mensagens entre os nós. No servidor do cluster deve estar o programa mestre e escravo, e nos nós escravos devem estar residentes os programas escravos [13]. O mestre é responsável pela otimização e execução do processo pelo escravo.

Um exemplo que utiliza o sistema PVM é o cluster Beowulf. Surgiu em 1994, nos laboratórios do Center of Excellence in Space Data Information Sciences – Centro de Excelência em Ciência da Informação e Dados Espaciais (CESDIS), tendo como finalidade processar as informações com baixo custo, podendo oferecer para o usuário um alto poder computacional.

Thomas Sterling e *Don Becker* foram os primeiros que construíram o cluster Beowulf com 16 nós de computadores, dotado de processadores do tipo 486 DX4-100 MHz, interligados através de uma rede Ethernet⁶ com 10 Mbits em topologia barramento, como mostra a Figura II, possibilitando o mesmo ser um cluster de alto desempenho computacional.

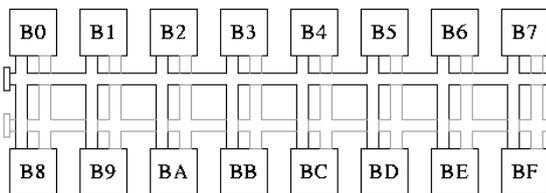


Figura III – Esquematização do primeiro cluster Beowulf em 1994 pelo CESDIS [14].

O Cluster Beowulf oferece várias vantagens em sua utilização e custo (possibilitando usar sistemas pessoais ou obsoletos) os quais são baratos para serem processados em conjunto e com alto desempenho, pois apresentam alta escalabilidade, utilizam o sistema PVM que é plataforma livre, ou seja, não precisa pagar licença para o seu uso e manutenção e possui alto desempenho, pois tem a capacidade de solucionar problemas de grande complexidade através do processamento paralelo, proporcionando uma diminuição no intervalo de tempo do seu processamento na resolução do problema [15].

Existem pesquisas de uso do PVM em redes de estações de trabalho em simuladores de reservatórios, efetuados por *Salazar, V. M.*, com o financiamento da Universidade de Campinas (UNICAMP) e Companhia de Petróleo Brasileiro S/A (PETROBRAS), onde se busca analisar o uso dos computadores que ficam ociosos no período noturno para fazer a análise de histórico dos reservatórios, gerando o parâmetro para análise de otimização da produção, calculando a diferença do real e o simulado, podendo ser usado também na análise de produção e cálculo de curvas na exploração de petróleo [16].

O PVM possui uma ferramenta de interface gráfica gratuita e monitoramento chamada XPVM, ilustrada na Figura III. Ela oferece várias visualizações animadas que auxiliam na depuração de erros e na otimização de desempenho do sistema[17].

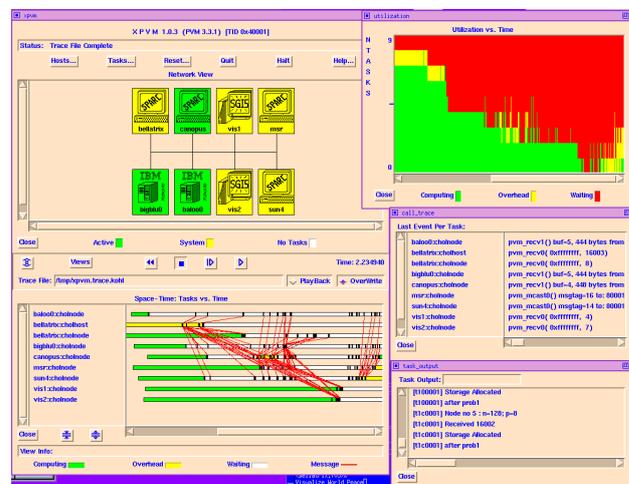


Figura IV – Interface do XPVM

Para analisar um programa usando XPVM, deve ser utilizado a biblioteca da versão 3.3 ou posterior, que foi instrumentado para capturar informações de rastreamento em tempo de execução[17].

Através da interface é possível efetuar alguns comandos do modo texto em modo interativo (*point-and-click*), como por exemplo adicionar ou remover servidores e escravos.

Além do XPVM existem outras ferramentas com as mesmas funcionalidades que podemos destacar: AIMS, PARADYN e PGPVM.

⁵ Heurística: Regras pré-definidas que reduzem o tempo de busca para determinada solução.

⁶ Ethernet: Tecnologia de interconexão de redes locais.

V. Comparativo entre PVM e outros sistemas de computação paralela.

Além do sistema PVM, temos outros sistemas no mercado como o *Windows HPC Server 2008*, que é um sistema proprietário desenvolvido pela *Microsoft* voltado para a computação de alta performance (High Personal Computing – HPC). Trabalha com vários nós de processamento, podendo escalar de dois a dois mil nós de servidores, sendo que cada nó pode ter entre um a quatro processadores. É totalmente integrado com as plataformas desktop da *Microsoft* como o *Microsoft SharePoint* e o *Microsoft Excel 2010*, permitindo o envio de tarefas a outros nós, como um documento para a fila de impressão.

É um sistema para computação em larga escala, onde se exige um alto poder de processamento como cálculos de risco de investimentos, de perfuração de campos de petróleo. Inclui ferramentas de gestão, um agendador de tarefas, *Microsoft Message Passing Interface* (MPI-MS), que permite que servidores diferente se comuniquem e trabalhem de forma natural, aplicação em ambiente de arquitetura orientada à serviços de planejamento (SOA).

A *Microsoft* oferece 5 versões do *Windows HPC Server 2008* [18]:

- *Windows HPC Server 2008 R2 Suite*: é a combinação do *Windows Server 2008 R2 HPC Edition* e do *Microsoft HPC Server 2008 R2 Enterprise Pack*. Proporciona um sistema operacional de 64 bits robusta para HPC e o middleware⁷ HPC necessário para implantar, gerenciar e desenvolver HPC habilitando as aplicações. Ambos os componentes estão disponíveis separadamente ou combinados como o *Windows HPC Server 2008 R2 Suite*.

- *Windows Server 2008 R2 HPC Edition*: é um sistema operacional de 64 bits do *Windows Server*, incluindo os recursos necessários e com a finalidade de executar aplicativos HPC agrupados.

- *Microsoft HPC Server 2008 R2 Enterprise Pack*: inclui um conjunto de funcionalidades incluídas no pacote de *HPC Server 2008 R2 Express*, juntamente com a capacidade de adicionar o *Windows 7* como nós de estação base e a capacidade de executar cálculos *Excel 2010* em todo o cluster usando um novo HPC Serviços.

- *Microsoft HPC Pack 2008 R2 Express*: inclui o software essencial para clusters HPC baseados em *Windows*, incluindo o agendamento de tarefas, passando interface de mensagens (MPI) e outros desenvolvimentos de interfaces de programação de aplicativos, administração de sistemas, implantação e suporte para redes de alta velocidade.

- *Microsoft HPC Pack 2008 R2 para estação de trabalho*: inclui o software para a implantação de sete estações de trabalho baseados em *Windows* como nós

⁷ Middleware: Programa que faz a integração entre microcomputadores que utilizam programas diferentes. Exemplo: Máquina virtual Java.

dentro de uma estação de trabalho de servidor baseadas em HPC cluster do *Windows*.

VI. Materiais e métodos

Foram realizados testes de carga com o software *POV-RAY* entre 3 equipamentos, que utiliza o sistema PVM para paralelização do processamento e divisão de tarefas.

O *POV-RAY* é um software livre multiplataforma usado na computação gráfica para criação de imagens em 3D. Utiliza a técnica de criação de imagens através de um modelo matemático da cena, levando em conta o usuário como centro de projeção (*raytracer*) para criação de imagens fotorealísticas em três dimensões [19].

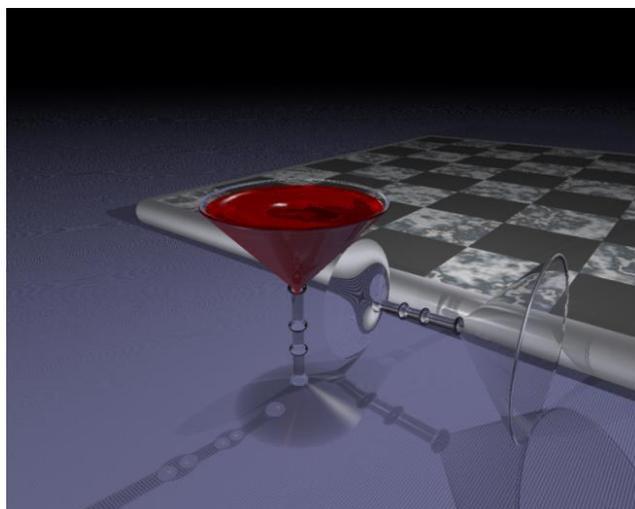


Figura V – Imagem gerada através do software *POV-RAY*, com o arquivo de texto *wg6.pov*

A imagem é gerada a partir das informações em um arquivo de texto contendo as descrições dos objetos, a iluminação e o ponto de vista da câmera. O arquivo de texto utilizado nos testes para gerar a imagem foi o *wg6.pov*, gerando a Figura V, na resolução de 1280x1024 pixels.

Três cenários diferentes foram simulados: Utilização de um único computador, dois computadores em paralelo e três computadores em paralelo. Em cada cenário, foi observado o tempo gasto para realizar a tarefa e a divisão das tarefas para cada equipamento.

A Tabela II descreve as configurações dos equipamentos utilizados.

Tabela II – Configurações dos equipamentos

Equipamento 1	Microcomputador com processador Intel Core 2 duo de 1.8 Ghz, memória RAM de 2 GB, HD de 40 GB, conexão de rede Ethernet de 100 Mbps, placa de vídeo onboard com 128 MB.
Equipamento 2	Microcomputador com processador Intel Atom de 1.6 Ghz, memória RAM de 1 GB, HD de 40 GB, conexão de rede Ethernet de 100 Mbps, placa de vídeo onboard com 64 MB.
Equipamento 3	Microcomputador com processador Intel Celeron de 2.2 Ghz, memória RAM de 1 GB e HD de 40 GB, conexão de rede Ethernet de 100 Mbps, placa de vídeo onboard com 64 MB.
Roteador	Equipamento para centralização e conexão entre os microcomputadores, com 4 portas Ethernet de 100 Mbps.
Cabos	Cabos de rede Ethernet categoria 5e

Foi utilizado um roteador para interligar os microcomputadores e foi utilizada a topologia de rede do tipo estrela, como mostra a Figura VI. O equipamento 1 foi classificado como servidor (mestre), pois possui a melhor configuração entre os 3 equipamentos.

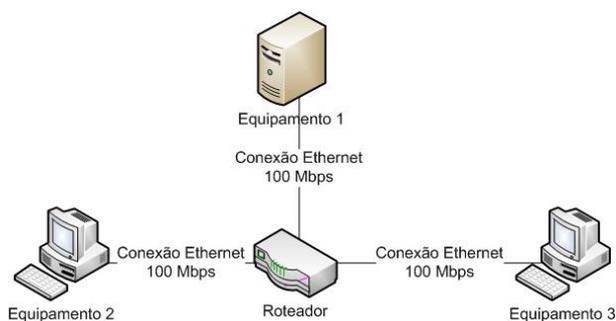


Figura VI – Topologia e disposição dos equipamentos na rede

Foram instalados nos 3 microcomputadores o sistema operacional Linux Ubuntu 10.10 (*Maverick Meerkat*) e na Tabela III estão descritos os pacotes (softwares, bibliotecas e dependências) instalados nos equipamentos.

Tabela III – Pacotes instalados nos equipamentos

SOFTWARE	DESCRIÇÃO TÉCNICA
PVM	Pacote para o sistema de máquina virtual paralela, versão 3.4.5-12.2
PVM-DEV	Pacote de desenvolvimento de softwares para máquina virtual paralela
XPVM	Gerenciador gráfico do PVM
OPENSSH-SERVER	Programa servidor para autenticação e comandos em máquinas virtuais.
OPENSSH-CLIENT	Programa cliente para autenticação e comandos em máquinas virtuais
POV-RAY	Programa de geração de imagens realísticas em 3D
PVMPOV	Programa de integração do software POV-RAY com PVM
GEANY	Programa para edição de textos avançado

Os testes foram divididos em 3 cenários:

Cenário 1: Execução do *POV-RAY* somente no equipamento 1.

Cenário 2: Execução do *POV-RAY* nos equipamentos 1 e 2 simultaneamente, utilizando o PVM e PVMPOV para distribuir a carga de processamento entre eles.

Cenário 3: Execução do *POV-RAY* nos equipamentos 1, 2 e 3, utilizando o PVM e PVMPOV para distribuir a carga de processamento entre eles.

Nos três cenários apresentados, foram monitorados o nível de processamento dos equipamentos envolvidos e o tempo gasto para execução.

Para obter um resultado mais preciso em relação ao tempo de execução, cada cenário foram executadas três vezes consecutivas e calculado a média entre os tempos obtidos.

Para o desempenho do processador, foi analisada a porcentagem de uso da CPU, utilizando o resultado mais freqüente no intervalo de tempo determinado no cenário.

VII. Resultados

No cenário 1, com a execução do *POV-RAY*, somente no equipamento 1, observou-se a utilização de 100% do processador e com o tempo médio de execução de 91,33 segundos, conforme Tabela IV.

Tabela IV – Relação de resultados no cenário 1

Testes	Processador (%)	Tempo (s)
1	100	90
2	100	92
3	100	92
Média	100	91,33

No cenário 2, com a execução do *POV-RAY* simultaneamente nos equipamentos 1 e 2, usou-se 81% do processador do equipamento 1 e 56% do equipamento 2 com tempo médio de 72 segundos, verificando que através do PVM, houve um melhor gerenciamento na distribuição da carga de processamento entre os microcomputadores e ganho médio de 19 segundos no processamento total.

A Tabela V demonstra os resultados obtidos no cenário 2.

Tabela V – Relação de resultados no cenário 2

Testes	Equipamento 1	Equipamento 2	Tempo (s)
	Processador (%)	Processador (%)	
1	81	56	72
2	83	55	68
3	79	58	76
Média	81	56	72

No cenário 3, com a inclusão do equipamento 3, usou-se em média 64% do processador do equipamento 1, 36 % do equipamento 2 e 43% do equipamento 3 com o tempo médio de processamento de 65 segundos.

A Tabela VI indica os resultados obtidos no cenário 3.

Tabela VI – Relação de resultados do cenário 3.

Testes	Equipamento 1	Equipamento 2	Equipamento 3	Tempo (s)
	Processador (%)	Processador (%)	Processador (%)	
1	64	40	45	61,5
2	66	31	40	70
3	62	37	44	63,5
Média	64	36	43	65

O Gráfico I demonstra a comparação entre o tempo de execução das tarefas com os cenários.

Observou-se que conforme o aumento na quantidade de processadores envolvidos na execução das tarefas, o tempo de execução diminuiu em média 20% para cada processador adicionado.

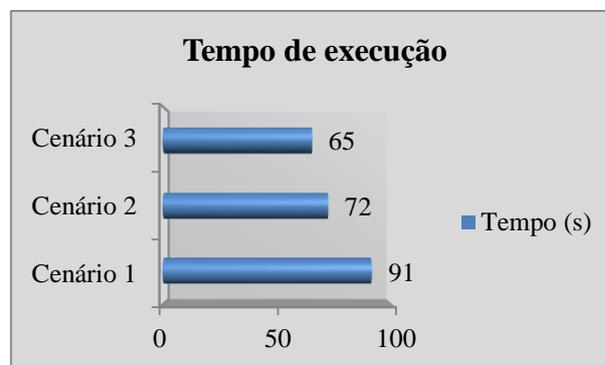


Gráfico I – Comparação entre o tempo de execução das tarefas e os cenários.

O Gráfico II descreve a quantidade de processamento utilizado em cada equipamento nos cenários.

Podemos observar que existe uma divisão da carga entre os processadores, mas não de forma uniforme. O equipamento 1, por ser o servidor, possui sempre um nível de processamento mais alto em todos os cenários analisados.

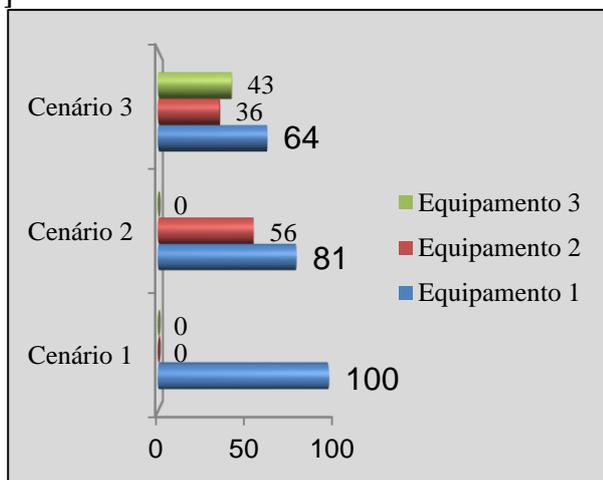


Gráfico II – Nível de processamento (em %)

VIII. Conclusão

O PVM permite o reaproveitamento dos microcomputadores para fins específicos como a renderização de imagens, diminuindo o tempo de execução conforme os resultados obtidos.

Mas, segundo Coqueiro [15], com sete equipamentos o tempo de processamento continua diminuindo, porém, nota-se que não há uma diminuição de tempo de execução considerável em relação a escalabilidade com cinco, seis e sete computadores, havendo pouca diferença entre eles.

O PVM pode ser uma alternativa para reciclagem de computadores

IX. Agradecimentos

Luciano Kazuo Tamehiro e Marcio Suekasu Takahashi agradecem ao Professor Ezequiel Ferreira dos Santos e a Professora Alice Flora Madeira Ribeiro pela orientação no artigo. Agradecemos aos nossos familiares e amigos, pela paciência e compreensão na elaboração deste artigo. Agradecemos a Universidade Nove de Julho por todos os materiais oferecidos através da biblioteca e dos professores do curso de Ciência da Computação.

X. Referências

[1] Elias, D., "Introduction to Parallel Programming Concepts", Workshop on Parallel Programming on the IBM SP, Cornell Theory Center, 1995.

Disponível através da WWW no endereço: <http://www.tc.cornell.edu/Edu/Workshop>, consultado no dia 07 de novembro de 2010.

[2] Santos, E. F., Souza, M. T. S., "Um estudo das motivações para implantação de programas de logística reversa de computadores", Revista eletrônica de ciência administrativa (RECADM) / Faculdade Cenecista de Campo Largo, Paraná, Brasil, 2009.

[3] Smith, P. H., "System software and tools for high performance computing environments", 1992. Disponível através da WWW no endereço <http://www.ccsf.caltech.edu/PSTP.html>, acessado em 08 de agosto de 2010.

[4] Rochman, N. T., Kawamoto, K., Sueyoshi, H., Nakamura, Y., Nishida T., "Effect of milling temperature and additive elements on an Fe-C system alloy prepared by mechanical alloying", Journal of Materials Processing Technology, Vol.89-90, p. 367-372, 1999.

[5] Site, <http://www.theapplemuseum.com/index.php?id=5>, acessado em 15/11/2010.

[6] Site, http://cache-www.intel.com/cd/00/00/40/17/401775_401775.pdf, acessado em 15/11/2010.

[7] Coulouris, G., Dollimore, J., Kindberg, T., "Distributed Systems: Concepts and Design", 3ª ed. Addison-Wesley, 2001. Disponível através da WWW no endereço <http://www.booksites.net/cdkbook>, acessado em 05 de novembro de 2010.

[8] Schiozer, D. J., "Computação Paralela Aplicada a Simulação Numérica de Reservatórios", Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Departamento de Engenharia de Petróleo, Campinas, 2003.

[9] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderman, V., "PVM: Parallel Virtual Machine – A user's guide and tutorial for networked parallel computing", MIT Press, Cambridge, 1994.

[10] Site, http://www.epm.ornl.gov/pvm/pvm_home.html, PVM, acessado em 15 de outubro de 2010.

[11] Site, <http://www3.di.uminho.pt/~amp/textos/pvm/node2.html#SECTION00236000000000000000>, acessado em 25 de outubro de 2010.

- [12] Amaral Feio, P. R. de C., “*Programação paralela utilizando cluster Beowulf e o sistema PVM*”, Instituto de Estudos Superiores da Amazônia - IESAM, Belém, 2005.
- [13] Kawasaki, R., “*Imagens fractais obtidas por processamento paralelo usando um cluster Beowulf*”, Instituto de estudos superiores da Amazônia, Belém, 2005.
- [14] Site, <http://www.inf.ufrgs.br/gppd/disc/cmp134/trabs/T1/981/BWulf/CMP-134-T1-3.html>, acessado no dia 26 de outubro de 2010.
- [15] Coqueiro, T. S., “*Supercomputadores de baixo custo financeiro: Implementação do Cluster Beowulf*”, Instituto de Estudos Superiores da Amazônia – IESAM, Belém, 2005.
- [16] Salazar, V. M., Schiozer, D. J., Monticelli, A. J., “*External Parallelization of Reservoir Simulators Using a Network of Workstations and PVM.*”, UNICAMP, Campinas, 1995.
- [17] Site, <http://www.netlib.org/utk/icl/xpvm/xpvm.html>, XPVM acessado em 16 de outubro de 2010.
- [18] Site, <http://www.microsoft.com/hpc/en/us/how-to-buy/volume-licensing.aspx>, acessado em 02 de novembro de 2010.
- [19] Site, <http://www.povray.org/documentation/view/3.6.1/3/>, acessado em 08 de novembro de 2010.