

O Processador de Alto Desempenho do CTI Um Estudo de Viabilidade

*Carlos Antônio Ruggiero*¹ *Luís Sérgio Vicira Dutra*²

Universidade de São Paulo
Departamento de Física e Ciência dos Materiais
Caixa Postal 369, São Carlos, São Paulo
CEP 13560-970 Tel: (0162) 74-9199
e
Centro Tecnológico de Informática (CTI)
Instituto de Computação
Rodovia Dom Pedro I, km. 143.6
CEP 13089-500, Campinas, São Paulo

RESUMO

Neste artigo apresentamos o estudo feito no Instituto de Computação do Centro Tecnológico de Informática para se determinar a melhor arquitetura para o novo processador de alto desempenho a ser construído nesse centro. Quatro arquiteturas de sistema são apresentadas e comparadas utilizando-se tanto critérios técnicos quanto de mercado.

ABSTRACT

This paper tries to establish the best architecture to be used in the next high performance computer to be developed at the "Instituto de Computação" of the "Centro Tecnológico para Informática". Four system architectures are presented. They are compared using both technical and commercial criteria.

¹e-mail: toto@ifqsc.usp.br

²e-mail: dutra@cti.usp.br

1 Introdução

Máquinas computacionais que apresentam grande poder de processamento são conhecidas pelo nome genérico de *supercomputadores*. A utilização dessas máquinas tem crescido bastante nos últimos 15 anos abrangendo áreas como processamento de dados sísmicos, processamento de imagens médicas, análise estrutural, simulação de circuitos analógicos entre outras [1]. Não se trata de apenas um computador mais rápido para executar aplicações que, de qualquer forma, seriam executadas (embora mais lentamente) por minicomputadores ou "mainframes"; supercomputadores são as únicas alternativas viáveis para a resolução de certos problemas onde a demora pode tornar inútil o resultado (um exemplo típico é a meteorologia). Além disso, a rapidez na obtenção da solução desejada pode representar uma vantagem competitiva considerável em áreas onde o lançamento de um produto no mercado deva ser feito o mais cedo possível.

Pelo comportamento do mercado de computação na década de 80, é provável que, nos anos 90, a computação científica se abra em duas áreas principais: a área de computadores pessoais incluindo máquinas do tipo "PC" e estações de trabalho e a de supercomputadores tais como as grandes máquinas vetoriais da Cray [2], NEC e IBM. Esse fato é sugerido pela grande queda na venda dos minicomputadores nos últimos anos no mercado americano. Por outro lado, usuários que realmente precisam de alto poder computacional estão cada vez mais dispostos a pagar os altos preços necessários para a compra e manutenção de supercomputadores e esse mercado tem crescido muito.

Além de sua importância em áreas fundamentais como previsão de tempo e processamento de imagens, outro motivo faz com que seja fundamental o domínio da tecnologia de supercomputadores pelo Brasil: a dificuldade que se tem encontrado para importar máquinas realmente rápidas. De fato, os países que produzem supercomputadores, nitidamente EUA e Japão, têm imposto severas restrições à venda de tais equipamentos para o Brasil.

É importante lembrar porém, que o próprio conceito de supercomputador tem mudado nos últimos anos. A princípio se consideravam supercomputadores somente as grandes máquinas vetoriais feitas com tecnologia de ponta tais como o Cray-1 e Cyber 205 [3]. Atualmente, porém, acredita-se que somente a exploração do paralelismo pode levar à construção de computadores mais rápidos [4] de forma que, além de possuírem unidades vetoriais, supercomputadores devem ser capazes de executar várias atividades simultaneamente fazendo uso de várias unidades funcionais. Acredita-se que, pela exploração de paralelismo, consigam-se velocidades de processamento maiores com um custo mais baixo melhorando a relação custo desempenho dos supercomputadores.

Esse documento apresenta um estudo de viabilidade de um processador de alto desempenho (PAD) a ser desenvolvido no Centro Tecnológico de Informática (CTI). Serão analisadas e comparadas várias alternativas de arquitetura de sistema para o PAD.

2 Objetivos

O objetivo primeiro do projeto PAD é produzir uma máquina com poder computacional muito mais alto do que aquele encontrado em computadores pessoais e estações de trabalho mas com preço acessível. Como parâmetros iniciais, acredita-se poder obter um desempenho compatível com aquele dos minisupercomputadores atuais por um preço final não superior a 300 mil dólares americanos. Apesar do preço, e como pode ser deduzido da introdução, não é intenção do CTI construir um minicomputador mas sim um computador científico de alto desempenho.

Antes de propormos algumas arquiteturas possíveis para o PAD, devemos estabelecer alguns critérios para que as propostas sejam avaliadas. Estes critérios, que se relacionam com o que se deseja do PAD são:

- **Custo do Desenvolvimento e do Produto:** Deseja-se que o projeto PAD tenha custo mínimo pois os recursos disponíveis (equipamento, material de consumo e recursos humanos) são limitados. Na verdade, é esse o motivo para não nos propormos a construir um supercomputador do porte dos Crays e NECs. O custo do produto não deve ser muito alto pois pretende-se que a máquina seja usada em várias instituições nacionais que não teriam recursos para financiá-la.
- **Impacto no Mercado:** O CTI não atua diretamente no mercado mas espera-se que o PAD seja de interesse para vários setores no Brasil, incluindo instituições de ensino superior tais como Universidades e centros de pesquisa e empresas usuárias ou que venham a comercializar o produto.
- **Facilidade de utilização:** É fundamental que a máquina seja de fácil utilização e programação para que possa ser utilizada com conforto por uma grande gama de usuários. Este fator é de particular importância na exploração de paralelismo pois a experiência tem demonstrado que de nada adianta uma máquina com enorme potencial de desempenho se sua programação requer um grande esforço de especialistas em sua arquitetura.
- **Confiabilidade:** Todo protótipo que pretenda se tornar um produto deve apresentar alto grau de confiabilidade. É importante ter em mente que sistemas muito complexos, com muitos componentes e com grande dissipação de calor tendem a ser pouco confiáveis. Assim pode ser necessário sacrificar o desempenho para aumentar a confiabilidade.
- **Desempenho:** É claro que é muito importante maximizar-se o desempenho da máquina mas deve-se lembrar que, muitas vezes, um alto desempenho implica em dificuldade de utilização, pouca confiabilidade do produto e alto custo, todas características indesejáveis. Acreditamos que é fundamental a exploração de paralelismo para se conseguir alta velocidade de processamento a baixo custo.

3 Identificação do Problema e Soluções Propostas

Pretende-se que o PAD seja uma máquina de alto desempenho em aplicações numericamente intensivas. A computação numericamente intensiva é de grande importância em diversas áreas tais como prospecção de petróleo, previsão de tempo, cálculo de estruturas, processamento de imagens médicas entre outras [1]. Além disso, o CTI já apresenta considerável conhecimento na área, haja visto o projeto recém concluído do Coprocessador de Ponto Flutuante (CPPF), o que diminui a necessidade de treinamento do pessoal de projeto reduzindo custos e tempo de desenvolvimento.

Para que o PAD não se transformasse num exercício puramente acadêmico, buscou-se a Universidade de São Paulo (Instituto de Física e Química de São Carlos) para se encontrar um “cliente” com necessidades reais de computação numérica. Decidiu-se então, que as duas aplicações alvo seriam processamento de sinais para ressonância magnética nuclear e cálculos de dinâmica molecular.

É importante lembrar que, apesar do PAD ter como objetivo executar essas aplicações o mais eficientemente possível, ele será uma máquina de propósito geral, e, dessa forma, deverá apresentar alto desempenho também em outras aplicações numericamente intensivas.

Considerando-se os objetivos do PAD e as aplicações alvo a que se destina, 4 modelos de sistemas computacionais podem ser considerados para virem a ser o modelo adotado. Esses modelos são expostos nas subseções que se seguem.

3.1 Modelo Cliente-Servidor

Nesse modelo, várias estações de trabalho executando o sistema operacional Unix cooperam na solução do problema através de uma rede local de dados. As estações todas devem possuir o “Remote Procedure Call” (RPC) (na verdade, nem é necessário que todas executem Unix desde que possuam o RPC). Como atualmente, praticamente todas as estações do mercado já vem equipadas com o mecanismo RPC, o problema aqui se reduziria ao da programação das aplicações desejadas.

3.2 Modelo Heterogêneo

Nesse modelo, tem-se um computador principal do porte de um minicomputador como o VAX da Digital americana como hospedeiro e um coprocessador fracamente acoplado (*loosely coupled*) com processamento vetorial que é o responsável pelo processamento numérico intensivo. A interface com o usuário é feita através de uma estação de trabalho ou de um terminal X (que emula Xwindows) ligada ao minicomputador. O modelo é chamado de “heterogêneo” por conter três arquiteturas que, individualmente, são utilizadas em áreas completamente diferentes. Deve ser prevista a possibilidade de se utilizar mais de um coprocessador

para acelerar o processamento. O hospedeiro será o responsável pela distribuição de trabalho entre os vários coprocessadores.

3.3 Modelo Amoeba

A idéia aqui é parecida com a do modelo cliente-servidor só que, ao invés das estações executarem o sistema Unix, elas executam o sistema distribuído Amoeba idealizado por Andrew Tanenbaum e outros. Esse sistema é otimizado para a execução de aplicações distribuídas e contém no seu núcleo as primitivas necessárias para comunicação entre processos. A figura 1 mostra a arquitetura dos sistema

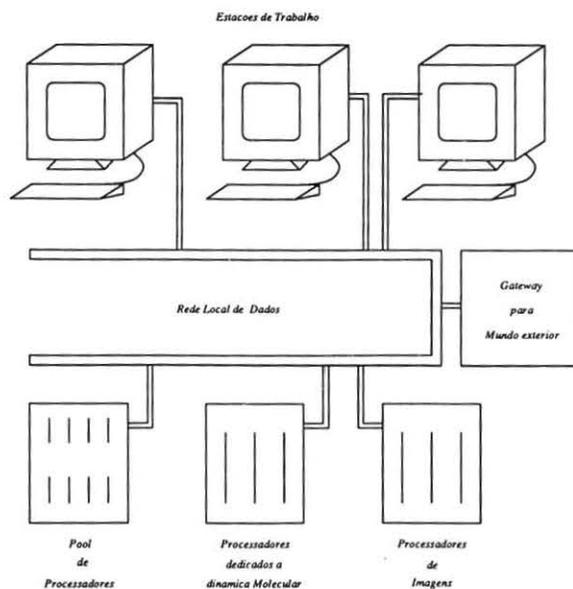


Figura 1: Modelo Amoeba

Note que nesse modelo, tem-se conjuntos de processadores dedicados ligados à rede. No caso do PAD, ter-se-iam processadores dedicados ao processamento de Dinâmica Molecular e ao processamento de imagens.

Além do Amoeba, existe uma variedade de sistemas operacionais distribuídos que poderiam ser utilizados tais como o 5, o Chorus, entre outros.

3.4 Modelo Vetorial Paralelo

Esse modelo é parecido com o heterogeneo só que aqui, o acoplamento entre o hospedeiro e o coprocessador vetorial é forte (via barramento) de tal forma que se torna agora necessário projetar o sistema como um todo incluindo processamento hospedeiro, processamento gráfico e processamento numericamente intensivo. Um único módulo conterá o processador escalar (central), um coprocessador para processamento numérico e um processador de comunicações. Num segundo nível, vários módulos de processamento podem ser interligados através do processador de comunicações para fornecer alto desempenho. A figura 2 mostra a arquitetura do sistema.

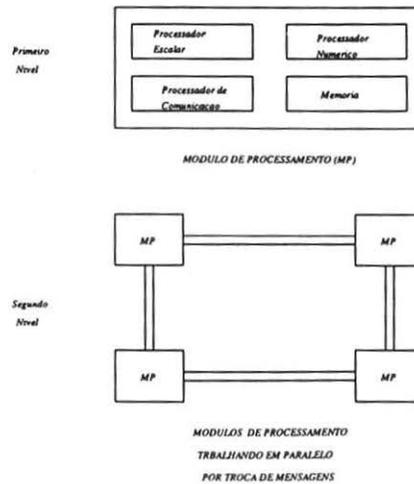


Figura 2: O Modelo Vetorial Paralelo

Deve-se utilizar como processador central, uma pastilha de microprocessador para o qual já houvesse uma grande base instalada. Isso eliminaria a necessidade de desenvolvimento de software básico dentro do CTI reduzindo consideravelmente o custo do projeto. Possíveis escolhas para o microprocessador são o SPARC da Sun Microsystems, o Alpha da DEC e o 586 da Intel. O projeto do módulo de processamento já incluiria – ou pelo menos levaria em conta – o projeto do coprocessador para processamento numérico.

4 Análise das Soluções Propostas

4.1 Introdução

O projeto do PAD pretende chegar a uma máquina de alto desempenho com características comerciais como ficou claro na introdução. É importante portanto, avaliar-se os vários modelos de acordo com os critérios mais utilizadas pelas indústrias de computação. Nesse documento escolhemos os seguintes elementos para a comparação dos modelos propostos: custo do projeto, custo do produto, dependência tecnológica (vida útil do modelo), marketing tecnológico, engenharia do produto, escalabilidade, velocidade numa aplicação, "throughput", facilidade de migração de programas sem que o desempenho máximo seja atingido e facilidade de migração de programas aproveitando bem o "hardware".

É óbvio que o *custo do projeto*, assim como o custo do produto deve ser o mais baixo possível. Isso é particularmente verdadeiro na atual situação econômica brasileira já que os recursos para financiamento são muito baixos. Nessa situação, um alto custo de desenvolvimento pode implicar na interrupção abrupta do projeto antes de se atingir a meta final.

Para definirmos o *custo do produto* é preciso definir o porte da máquina desejada. Para efeito de comparação, decidimos fixar o preço total do sistema (preço final) em 300.000 dólares americanos e calcular aproximadamente a configuração de cada um dos 4 modelos sugeridos. Dessa forma, esse item torna-se constante para todos os modelos.

Num projeto com horizonte de alguns anos como o PAD, é fundamental usar-se uma tecnologia que se mantenha atual durante esse período. Devido à alta velocidade de mudança da indústria de informática, esse objetivo é difícil de ser alcançado. Porém, é fácil ver que os modelos apresentados diferem quanto à sua habilidade em absorver os avanços tecnológicos e assim introduzimos o conceito de *dependência tecnológica*.

Apesar de ser desenvolvido num centro de pesquisa, o PAD pretende ter as características de um produto comercial. Assim, é importante que o produto seja atraente para os futuros usuários. Esse item foi considerado em nossa análise sob o título de *marketing tecnológico*. É importante notar que, alguns produtos na área de informática, incluindo alguns de alta qualidade técnica não foram bem sucedidos comercialmente por falhas de "marketing".

O item de avaliação *engenharia do produto* analisa fatores tais como tempo médio entre falhas do PAD, replicabilidade, empacotamento e consumo de potencia. Esse item representa a preocupação que se deve ter para que o protótipo possa ser facilmente transformado em produto comercial e que tem sido regularmente esquecida em muitos projetos nacionais na área de informática.

Por *escalabilidade* entende-se a habilidade de um sistema multiprocessador ter seu desempenho aumentado com a introdução de mais processadores. Esse aumento de desempenho deve, idealmente, ser

conseguido sem qualquer modificação no software e com poucas modificações no hardware. É fundamental escolher um modelo computacional que apresente tal propriedade para garantir que velocidades altas sejam atingidas se o usuário estiver disposto a pagar mais por processadores extras.

A *velocidade numa aplicação* caracteriza o potencial de desempenho dos vários modelos na execução das aplicações alvo do PAD, a dinâmica molecular e o processamento de imagens. É importante que o protótipo apresente um desempenho nas aplicações alvo maior do que o das máquinas hoje disponíveis para os usuários dessas aplicações no Brasil.

O *throughput* é a capacidade do modelo de executar várias tarefas simultaneamente com bom desempenho. Apesar de ser uma característica interessante para um processador de propósito geral, certamente é menos importante que a velocidade numa aplicação única no caso do PAD.

Toda máquina nova deve permitir aos programadores migrarem facilmente os programas já existentes para a nova plataforma. Idealmente, essa migração deve produzir um programa que aproveite todo o potencial do novo hardware e essa característica é por nós chamada de *Migração com Desempenho*. Mesmo que o programa resultante não apresente um desempenho excelente, é importante garantir que ele execute corretamente na nova plataforma. Essa característica é especificada por *Migração sem Desempenho*.

Nas seções subsequentes, analisamos os vários modelos propostos de acordo com esses critérios tentando estabelecer o mais adequado para o PAD.

4.2 Modelo Cliente Servidor

Com o critério *custo do produto* fixo em 300.000 dólares, é preciso determinar-se qual a configuração do sistema para se avaliar fatores tais como *desempenho*, *engenharia de produto*, *throughput* entre outros. Consideremos uma estação básica como sendo composta por um módulo de processamento com 16 megabytes de memória principal e capacidade de processamento de 5 megaflops, controlador de rede, terminal de vídeo gráfico colorido, e subsistema de disco de 600 megabytes. Um preço aproximado para essas estações é de 20.000 dólares (é claro que este preço varia dependendo do fabricante; o valor dado está próximo do mínimo possível). Portanto seria possível ter-se 15 estações no sistema.

A figura 3 mostra a configuração sugerida.

Quanto aos elementos de análise da seção anterior, pode-se dizer que:

1. **Custos do Projeto:** É claro que esse modelo apresenta os menores custos pois aqui, só estão envolvidos os custos de software já que o hardware é composto por estações comerciais que seriam compradas. O tempo de desenvolvimento também será mínimo.

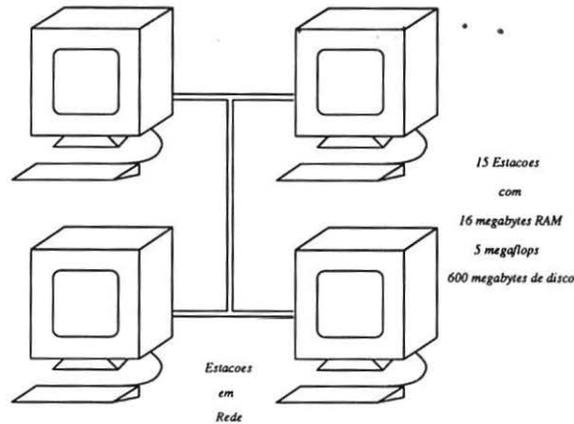


Figura 3: Configuração do sistema Cliente Servidor

2. **Dependência Tecnológica:** Outra característica positiva do modelo já que, com o avanço da tecnologia, é sempre possível substituir as estações por outras com maior desempenho sem qualquer alteração no software. A própria rede que interliga as estações pode ser substituída por outra de maior desempenho (por exemplo, de ethernet para FDDI) sem que isso altere o projeto básico.
3. **Marketing Tecnológico:** Aparentemente seria difícil convencer usuários de computação numericamente intensiva que um sistema composto por um conjunto de estações comerciais resolveria seus problemas eficientemente. De fato, nesse mercado, os usuários exigem desempenho elevado, difícil de ser conseguido com esse modelo (veja item *Velocidade numa Aplicação* abaixo).
4. **Engenharia de Produto:** Novamente, esse item não apresenta maiores dificuldades para o modelo devido à sua simplicidade de concepção.
5. **Velocidade numa aplicação:** Esse é talvez a maior desvantagem do modelo. De fato, se supusermos máxima eficiência na execução de um determinado algoritmo de forma que todas as estações recebam a mesma carga computacional e não fiquem jamais ociosas, o desempenho máximo possível será de $15 \times 5 = 60$ megaflops. Atualmente, esse desempenho seria considerado fraco para um sistema que se pretende aplicável à computação numericamente intensiva. Além disso, a dificuldade de se obter tal desempenho seria enorme pois a granularidade da aplicação deveria ser muito alta com pouca comunicação entre os processos. Como ilustração, suponhamos que as estações estejam ligadas através de uma rede ethernet. Duas estações dificilmente conseguiriam se comunicar a uma razão maior que 300 kbytes por segundo o que significa que, no máximo, um dado de 64 bits pode ser transmitido a cada:

$$\frac{8}{300 \times 10^3} = 2.6 \times 10^{-5} \text{segundos} = 26 \text{microsegundos} \quad (1)$$

Assim, para cada dado transmitido, seria necessário que a estação tivesse executado pelo menos 130 instruções de ponto flutuante ($26\mu s/200ns$) ou então não atingiríamos os 5 megaflops por estação. Mesmo esse cálculo é muito otimista pois supõe a inexistência de colisões na rede o que é claramente absurdo já que teríamos 15 estações tentando se comunicar a 300 kbytes por segundo sobrecarregando a rede. Porém, impor que não haja comunicação numa razão maior que um dado a cada 130 operações é claramente limitante para muitos algoritmos importantes inclusive para aqueles utilizados nas nossas aplicações principais, a dinâmica molecular e o processamento de imagem.

6. **Escalabilidade:** A adição de mais estações à rede dificilmente aumentaria o desempenho na maioria das aplicações já que a rede rapidamente atingiria a saturação e a velocidade total seria limitada pela velocidade da rede.
7. **Thruput:** A quantidade total de trabalho que um sistema de estações pode executar é bastante alta. Em nosso caso, o sistema poderia servir, simultaneamente, a 15 usuários (no mínimo) executando diferentes aplicações (numéricas, gráficas, etc).
8. **Facilidade de Migração de Software sem Desempenho:** Desde que sejam escolhidas estações de trabalho comerciais com grande base instalada, a portabilidade dos programas existentes estaria garantida.
9. **Facilidade de Migração de Software com Desempenho:** Para se obter um alto desempenho de uma aplicação, seria fundamental programá-la utilizando-se da filosofia cliente-servidor. Também seria importante garantir uma distribuição de carga computacional igualitária entre as estações. Isso envolveria um alto custo tanto em pessoal quanto em tempo de desenvolvimento.

4.3 Modelo Heterogêneo

Nesse modelo devemos ter um minicomputador que poderá ser configurado da seguinte forma: um coprocessador vetorial, 128 megabytes de memória principal, subsistema de disco de 10 gigabytes e 8 terminais X (ou seja, terminais capazes de visualização gráfica obedecendo ao protocolo "X windows"). Esse sistema seria responsável pelo processamento principal reservando o processador paralelo para aplicações numericamente intensivas. O custo aproximado dessa configuração estaria em torno de 200.000 dólares o que deixaria 100.000 dólares para o processador paralelo. Não é simples calcular exatamente a quantidade de recursos que seria possível incluir no processador paralelo com o limite de 100.000 dólares para o preço final sem ter-se um projeto detalhado. Assim temos que recorrer a um cálculo aproximado que será exposto a seguir. Para o preço final ser 100.000 dólares, deveríamos ter um custo total de componentes de aproximadamente 20.000 dólares (um quinto do valor final). Com isso seria possível construir um sistema multiprocessamento contendo por volta de 10 processadores rápidos do tipo por exemplo da "sparc" da Sun Microsystems ou do "alpha" da DEC. A configuração do sistema para avaliação pode ser vista na figura 4.

Quanto aos elementos de análise pode-se dizer que:

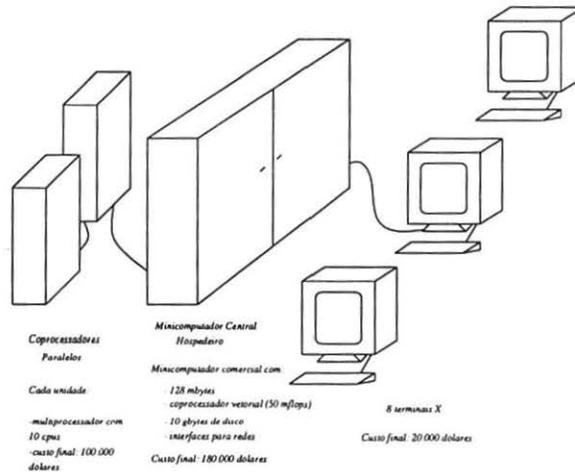


Figura 4: Configuração básica do sistema heterogêneo

1. **Custos do Projeto:** Claramente o custo aqui é mais alto do que no caso anterior do modelo cliente servidor, pois, agora, existe um projeto completo de hardware: o multiprocessador paralelo. Além disso, o trabalho em software também seria considerável já que além da programação das aplicações alvo, seria interessante fornecer uma ferramenta de paralelização de código para um melhor aproveitamento do processador paralelo. Esse trabalho certamente não é trivial.
2. **Dependência Tecnológica:** Aqui temos um problema sério do modelo já que é reconhecido atualmente que minicomputadores representam uma porcentagem cada vez menor do mercado global da computação. De fato, só no ano de 1991, as vendas de minicomputadores caiu 39% nos Estados Unidos e as empresas da área estão passando por sérias dificuldades financeiras. Assim, qualquer que seja a escolha quanto à marca do minicomputador, é provável que em tempo muito curto esse minicomputador já esteja obsoleto comprometendo a vida útil do modelo.
3. **Marketing Tecnológico:** O mercado parece receber bem a idéia de um sistema heterogêneo com possibilidades gráficas, vetoriais e paralelas. O problema aqui também seria a utilização de um minicomputador como visto no item anterior.
4. **Engenharia de Produto:** Devido à maior complexidade do projeto de hardware, é fácil ver que o sistema heterogêneo apresentaria maiores problemas nesse item do que o modelo cliente servidor analisado na seção anterior.
5. **Velocidade numa aplicação:** É difícil avaliar-se exatamente qual seria o desempenho do sistema numa determinada aplicação sem termos o projeto detalhado do processador paralelo. Esse processador porém, seria certamente mais rápido do que a rede de estações do modelo cliente servidor já que a limitação da velocidade da rede aqui não seria um problema. Também, o processador ve-

torial incluído no minicomputador poderia fornecer um bom desempenho em aplicações facilmente vetorizáveis. Infelizmente porém, a ligação (fraca) entre o minicomputador central e o processador paralelo poderá limitar o desempenho do sistema em certas aplicações.

6. **Escalabilidade:** A adição de várias unidades de processamento paralelo certamente poderá aumentar o desempenho mas, a cada processador adicionado, a limitação da velocidade do barramento torna-se mais nítida.
7. **Throuput:** A quantidade total de trabalho que o sistema pode executar é alta desde que se escolha um minicomputador comercial otimizado para "time-sharing". O processador paralelo porém, ficará boa parte do tempo subutilizado se o sistema for utilizado como de propósito geral.
8. **Facilidade de Migração de Software sem Desempenho:** Com a escolha de um minicomputador popular como por exemplo o VAX da DEC, a portabilidade de software fica bastante facilitada.
9. **Facilidade de Migração de Software com Desempenho:** Não deverá ser muito difícil transferir programas altamente vetorizáveis para o sistema usando eficientemente o processador vetorial do computador hospedeiro (minicomputador). O uso eficiente do processador paralelo é bem mais difícil e dependerá muito das ferramentas de otimização e avaliação disponíveis aos programadores.

4.4 Modelo Amoeba

Suponhamos a seguinte configuração de um sistema amoeba com os respectivos preços: 8 estações de trabalho muito simples e sem disco só fornecendo habilidade gráfica (40.000 dólares), um servidor de arquivos com 10 gigabytes (70.000 dólares), um servidor de processamento (140.000 dólares) e um processador dedicado à dinâmica molecular e ao processamento de imagens (50.000 dólares). O preço dos dois últimos itens poderia ser variável com a soma ficando fixa em 190.000 dólares. A figura 5 mostra a configuração sugerida.

Quanto aos elementos de análise pode-se dizer que:

1. **Custos do Projeto:** O custo do projeto do sistema amoeba é certamente maior que o custo do sistema cliente servidor pois aqui, como no caso do sistema heterogeneo, é necessário o projeto de unidades de hardware tais como o "pool" de processadores e a unidade dedicada aos problemas alvo. É possível basear-se em trabalhos já executados no caso do "pool" de processadores mas a unidade dedicada deve ser projetada integralmente. O projeto na área de software também será oneroso em recursos humanos e em tempo.
2. **Dependência Tecnológica:** A vida útil do modelo amoeba parece que será longa pois ele foi desenvolvido recentemente já prevendo avanços tecnológicos futuros. Porém, apesar do sistema operacional amoeba ter claras vantagens sobre sistemas operacionais convencionais como o Unix, é duvidoso se

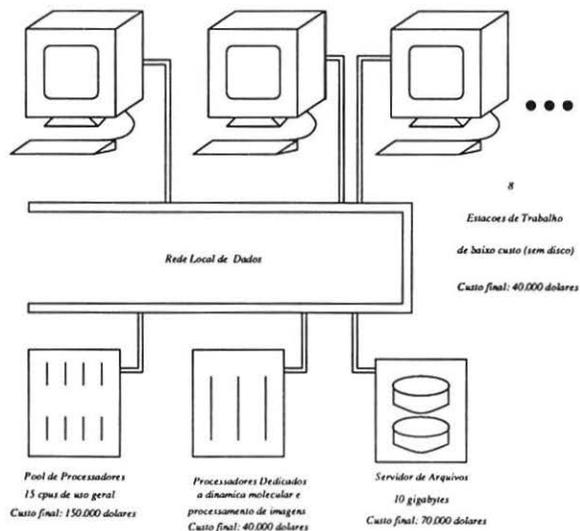


Figura 5: Configuração do Sistema Amoeba

realmente ele terá aceitação nos meios científicos e comerciais. Note que, se o sistema operacional amoeba não for aceito, todo o nosso sistema amoeba se tornará inviável.

3. **Marketing Tecnológico:** Devido ao fato de o amoeba ser um sistema operacional completamente novo, é natural que os usuários ofereçam alguma resistência à sua utilização. Esse é um sério problema desse modelo um pouco amenizado pela disponibilidade de um emulador Unix interno ao sistema.
4. **Engenharia de Produto:** Acreditamos que as dificuldades aqui sejam equivalentes às que seriam encontradas no modelo heterogeneo pois a complexidade de hardware é semelhante.
5. **Velocidade numa aplicação:** As aplicações alvo certamente seriam beneficiadas pela inclusão, em nossa configuração amoeba, de um processador dedicado a elas. Entretanto, a integração entre essa unidade dedicada e o restante do sistema poderia introduzir "overheads" que comprometeriam o desempenho.
6. **Escalabilidade:** É difícil prever exatamente o desempenho do sistema se fossem introduzidas unidades extras. Pela literatura existente sobre o amoeba é provável que a adição de novas unidades realmente produza grandes ganhos em desempenho sem envolver muito trabalho extra de configuração do hardware e do software.
7. **Throuput:** O throuput do sistema é muito bom apesar de que, é claro, num ambiente de muitos usuários é provavel que a unidade dedicada fique subutilizada.
8. **Facilidade de Migração de Software sem Desempenho:** O grande problema aqui é o fato de o sistema operacional amoeba ainda não ser aceito por uma quantidade considerável de usuários o que

torna difícil a instalação de programas já existentes.

9. **Facilidade de Migração de Software com Desempenho:** O sistema Amoeba foi todo desenvolvido levando-se em conta a necessidade de processamento paralelo para se atingir alto desempenho. Assim, ele parece ser o modelo mais apropriado, entre todos os apresentados, para execução eficiente de programas paralelizáveis. De fato, existe até uma linguagem chamada Orca especialmente voltada à exploração de paralelismo.

4.5 Modelo Vetorial Paralelo

A configuração desse modelo que será utilizada para efeito de comparação é composta de 8 terminais X no valor de 20.000 dólares, um conjunto de disco rígido com 10 gigabytes no valor de 30.000 dólares e 4 módulos de processamento (MP na figura 2) contidos no gabinete principal do sistema. Acreditamos ser possível a inclusão de 4 módulos de processamento com o preço final de 250.000 dólares, ou seja, 50.000 dólares para os componentes dos quatro módulos. A figura 6 ilustra essa configuração.

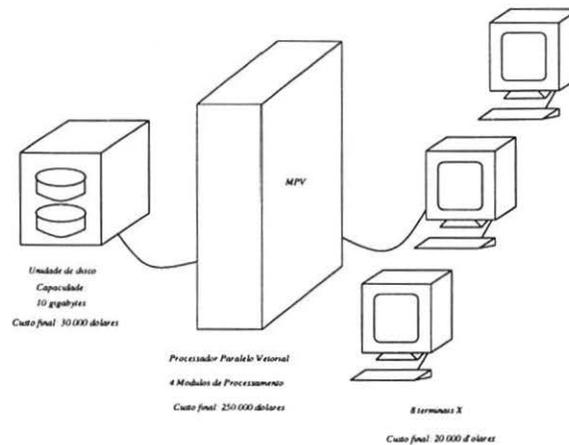


Figura 6: Configuração do sistema paralelo vetorial

Quanto aos elementos de análise pode-se dizer que:

1. **Custos do Projeto:** Devido à sua grande complexidade esse modelo apresenta o maior custo de projeto tanto em recursos humanos quanto em tempo de desenvolvimento.
2. **Dependência Tecnológica:** Se o processador escalar do módulo de processamento for bem escolhido, a vida útil do sistema deverá ser longa.
3. **Marketing Tecnológico:** Acreditamos que o mercado esteja bastante receptivo para máquinas capazes de explorar tanto o processamento vetorial quanto o paralelo.

4. **Engenharia de Produto:** Entre todos os modelos propostos, este e provavelmente o que apresentará maiores problemas na transformação em produto devido ao alto desempenho possível e à complexidade do projeto.
5. **Velocidade numa aplicação:** Como o projeto integral de toda a parte central de processamento deverá ser feita, é esperado que esse sistema possa apresentar o melhor desempenho pois a adaptação da arquitetura aos problemas alvo deverá ser boa.
6. **Escalabilidade:** É incerta a habilidade do modelo em apresentar ganhos consideráveis de desempenho com a introdução de módulos de processamento extras pois a distribuição igualitária de carga computacional entre processadores vetoriais é um problema de difícil solução.
7. **Throughput:** A quantidade total de trabalho realizado pelo sistema deverá ser boa apesar de que é inevitável que os processadores vetoriais fiquem ociosos boa parte do tempo.
8. **Facilidade de Migração de Software sem Desempenho:** Deverá ser muito boa desde que se escolha como processador escalar, uma pastilha de boa aceitação no mercado para a qual já exista uma grande quantidade de programas desenvolvidos.
9. **Facilidade de Migração de Software com Desempenho:** Como já foi dito anteriormente, o bom aproveitamento de vários processadores vetoriais não é uma tarefa simples. Porém, o problema é simplificado quando o número de módulos é pequeno como é o caso da configuração básica (4 módulos).

4.6 Escolha do Modelo

Das considerações feitas nas seções anteriores, é possível contruir uma tabela resumindo as vantagens e desvantagens de cada modelo (veja tabela 1). Para cada item, atribui-se um conceito B (bom), M (médio) ou R (ruim) para cada modelo. Em cada item de avaliação, pelo menos um modelo recebeu o conceito B.

Apesar de apresentar um bom conjunto de conceitos, o modelo cliente servidor deve ser eliminado pois apresenta um desempenho muito fraco nas aplicações alvo.

O modelo heterogeneo apresenta o sério problema de ser baseado em um minicomputador. O mercado de minicomputadores tem caído muito e isso coloca em risco a vida útil desse modelo. Também, é provável que a ligação fraca entre o minicomputador e os coprocessadores limite a velocidade máxima que o sistema pode obter.

Resta-nos então os dois últimos modelos. O modelo Amoeba seria certamente o melhor se o sistema operacional Amoeba fosse de larga aceitação. Infelizmente, porém, isso não ocorre tendo o Amoeba

<i>Ítem</i>	<i>Cliente Servidor</i>	<i>Heterogêneo</i>	<i>Amoeba</i>	<i>Paralelo Vetorial</i>
Custo do Projeto	B	M	M	R
Custo do Produto	B	B	B	B
Dependência Tecnológica	B	R	M	M
Marketing Tecnológico	R	M	R	B
Engenharia de Produto	B	M	M	R
Velocidade numa Aplicação	R	M	M	B
Escalabilidade	R	M	B	M
Throuput	B	M	M	M
Migração sem Desempenho	B	B	R	B
Migração com Desempenho	R	M	B	M

Tabela 1: Tabela comparativa entre os modelos

poucos adeptos na área de computação científica. O fato de que agora o sistema é vendido (e não dado como há alguns anos atrás) torna ainda mais difícil a sua popularização.

O modelo vetorial paralelo apresenta a grande vantagem de poder ter um desempenho excelente e ter grande apelo de mercado já que muitas empresas de supercomputação tais como a CRAY e a NEC tem projetado várias arquiteturas com essas características. Acreditamos assim, que esse é o melhor modelo para os propósitos do PAD apesar dos sérios problemas relacionados com a grande complexidade do projeto.

Referências

- [1] Hwang, K. & Briggs, F. A. *Computer Architecture and Parallel Processing*, McGraw-Hill, 1985.
- [2] Russel, R. M. *The CRAY1 Computer System*, Communications of the ACM, Vol. 21, No. 1, janeiro 1978.
- [3] Ibbet, R. N. *The Architecture of High Performance Computers*, The MacMillan Press, Ltd, 1982.
- [4] Almasi, G. S., Gottlieb, A. *Highly Parallel Computing*, The Benjamin/Cummings Publishing Company, 1989.