

INTRODUÇÃO AO PROCESSAMENTO PARALELO

Philippe O. A. Navaux
 CPGCC-DI-CPD-UFRGS
 CP 1501 - 90001, Porto Alegre, RS - BRASIL
 Tel.: (0512) 21.8499 Telex (051) 2680 CCUF

RESUMO

O presente trabalho apresenta a área de Processamento Paralelo nos seus principais tópicos desde as arquiteturas, linguagens, compiladores, até os algoritmos. São apresentadas classificações, terminologia empregada e aplicações, objetivando ressaltar os principais problemas que ainda precisam ser solucionados para um maior e melhor uso destas máquinas.

ABSTRACT

This paper will present an overview of Parallel Processing architectures, languages, compilers up to algorithms. Classifications, principal terms and applications will be presented and principal problems will be mentioned that need solutions for best usage of these machines.

1. INTRODUÇÃO

O que é Processamento Paralelo?

Segundo a definição de Hwang [2]: é uma forma eficiente do processamento da informação com ênfase na exploração de eventos concorrentes no processo computacional.

A idéia do Processamento Paralelo não é nova, em 1920 Vannevar Bush do MIT apresentava um computador analógico capaz de resolver equações diferenciais em paralelo; o próprio Von Neumann em seus artigos, pelos idos de 1940, sugere uma grade para resolver equações diferenciais em que os pontos são atualizados em paralelo. Porém é considerado como marco inicial no surgimento de máquinas para processamento paralelo o ILLIAC IV construído em fins de 60 na Universidade de Illinois composto de 64 processadores.

A razão comercial do surgimento do Processamento Paralelo é a capacidade de aumentar o processamento com uma única máquina. Como o aumento da velocidade nas máquinas seqüenciais é limitada pela tecnologia, a solução empregada para aumentar o poder de processamento é a utilização de processadores em paralelo. Claro que esta solução não é tão simples assim, pois além de existirem diversas formas de colocar os processadores em paralelo, surgem problemas na forma de como gerenciar estas máquinas, de como manter a coerência da informação, e outros itens que este artigo tentará abordar.

Conclui-se do que acima foi afirmado que processamento paralelo existe a partir do momento em que dois ou mais processadores interagem entre si para resolverem uma determinada tarefa. Portanto este assunto é muito vasto e implica desde algoritmos que possam ser paralelizados, até as arquiteturas paralelas. É importante salientar que paralelismo não implica necessariamente em máquinas de alta performance, conhecidos pelos SUPERCOMPUTADORES, mas trata também do paralelismo em situações de baixa velocidade; claro que usualmente quando se fala de paralelismo es-

tá tratando-se do primeiro caso.

O mercado destas máquinas está crescendo rapidamente devendo chegar a U\$ 6 bilhões (dólares) em 1990 segundo Tim Johnson [3]. Este crescimento que anda numa faixa de 35% ao ano representa uma das taxas mais altas do mercado. Diversas empresas estão surgindo nesta área, algumas já com alguma tradição, tais como a Cray, outras recentes, Convex Computer, Alliant Computer, Cydrome, Scientific Computer Systems entre outras. Esta facilidade no surgimento de novas empresas é em parte devido ao fato de que as grandes empresas, tais como IBM e DEC, ainda não atuam nesta área. Porém, considerando a expansão do mercado, não vai demorar a entrada destas grandes companhias; por exemplo, Steve Chen, que saiu da CRAY em setembro passado, está montando uma empresa [9], com apoio da IBM, para desenvolver uma máquina 100 vezes mais rápida que as atualmente disponíveis no mercado.

Portanto o Processamento Paralelo trata desde a Aplicação; a partição de algoritmos paralelos, as Linguagens Paralelas; os Compiladores Paralelos, compiladores vetorizadores, compiladores otimizadores; os Sistemas Operacionais para gerenciar este paralelismo; as Arquiteturas Paralelas; o Mapeamento de Algoritmos; e outros itens. O presente levantamento tentará, abordar alguns destes assuntos, apresentando o estado atual das pesquisas na área, ao longo das próximas seções. É importante salientar que tendo em vista ser uma área de pesquisa de ponta, muitos assuntos ainda não estão claros, assim como muitos termos ainda não possuem um significado aceito universalmente, portanto tentaremos seguir, da melhor maneira possível, o que é consenso na área, porém em alguns momentos deveremos assumir alguns termos ou definições para facilitar a clareza do artigo, muito embora estas ainda não sejam consenso.

2. CLASSIFICAÇÕES E UNIDADES DE MEDIDAS DO PROCESSAMENTO PARALELO

Com vistas a estabelecer as características de contorno da área serão apresentados nos próximos itens algumas classificações e unidades de medida.

2.1. Classificação de Arquiteturas segundo Flynn

Flynn em 1966 [4] propôs uma classificação das arquiteturas em quatro categorias de máquinas conforme a multiplicidade do fluxo de dados e de instruções. Por ser bastante simples, esta classificação é a melhor aceita pela comunidade científica. Segundo esta o ponto principal do processo de um computador é a execução de um conjunto de instruções sobre um conjunto de dados. A palavra fluxo é empregada para descrever uma sequência de instruções, ou dados, executados num único processador. Portanto, um fluxo de instruções é uma sequência de instruções executadas por uma máquina, enquanto que um fluxo de dados é uma sequência de dados, de entrada, de resultados parciais ou intermediários, utilizados pelo fluxo de instruções [2] [5]. As quatro categorias de Flynn são: SISR (Single Instruction Stream - Single Data Stream), SIMD (Single Instruction Stream - Multiple Data Stream), MISD (Multiple Instruction Stream - Single Data Stream), MIMD (Multiple Instruction Stream - Multiple Data Stream).

A categoria SISR (fig. 1), fluxo de instruções e dados únicos, trata dos computadores comumente disponíveis na atualidade. São as máquinas baseadas nos princípios de Von Neumann. As instruções são executadas sequencialmente, mas podem ser sobrepostas nos seus estágios de execução (pipeline). Os computadores SISR podem ter mais de uma unidade funcional porém todos sob a supervisão de uma unidade de controle.

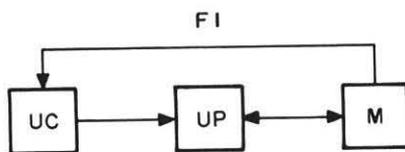


Fig. 1 Computadores do Tipo SISR

A categoria SIMD (fig. 2), fluxo de instruções único com vários fluxos de dados, corresponde aos processadores matriciais (array) paralelos e associativos, também chamados de arranjos paralelos e associativos. São vários elementos de processamento supervisionados por uma mesma unidade de controle, isto é, todos os elementos de processamento recebem a mesma instrução para operarem sobre diferentes dados. A memória do sistema pode estar dividida em módulos, e vinculados a cada elemento de processamento.

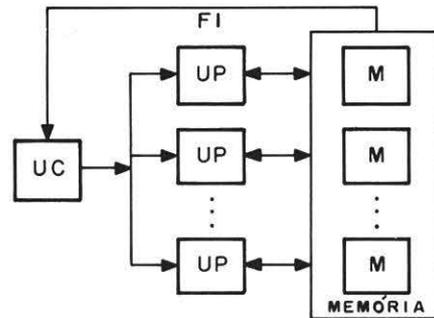
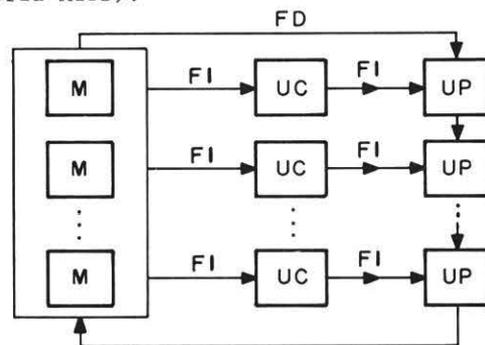


Fig. 2 Computadores do Tipo SIMD

A categoria MISD (fig. 3), múltiplos fluxos de instruções para um único fluxo de dados, corresponderia a uma máquina que tivesse vários elementos de processamento recebendo instruções distintas das unidades de controle, porém processando o mesmo fluxo de dados. O resultado de um processador seria a entrada do próximo. Esta estrutura não existe na prática, sendo inclusive considerada sem validade por alguns pesquisadores da área (alguns autores consideram o pipeline como representante da categoria MISD).



FI - Fluxo de Instruções M - Memória
FD - Fluxo de Dados UC - Unidade de Controle
UP - Unidade de Processamento

Fig. 3 Computadores do Tipo MISD

Na categoria MIMD (fig. 4), múltiplos fluxos de instruções e múltiplos fluxos de dados, são representativos os Multiprocessadores. Cada Unidade de Processamento, possui sua Unidade de Controle executando instruções sobre um conjunto de dados; e o que mantém a interação entre os diversos processadores é a memória. Se tal não acontecesse estaríamos frente a um sistema de múltiplos SISR (MSISR). Os MIMD podem ser fracamente acoplados ou fortemente acoplados dependendo do grau de interação menor ou maior existente entre os vários processadores através da memória. A memória que executa este elo de ligação entre processadores é conhecido como memória global. Os poucos MIMD existentes comercialmente possuem características do tipo multiprocessadores fracamente acoplados.

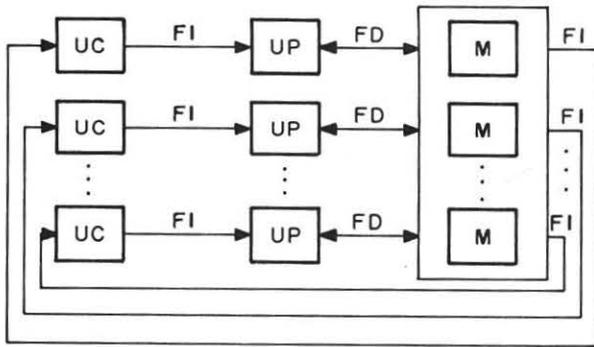


Fig. 4 Computadores MIMD

2.2. Unidades de Medida para Máquinas Paralelas

Para medir a velocidade das Máquinas Paralelas foi estabelecido uma unidade: o FLOP. Assim como os MIPS representam o milhão de instruções executadas num segundo por uma máquina, o MEGAFLOPS (ou MFLOPS) mede o número de milhões de operações em ponto flutuante executado num segundo. Não existe uma conversão fixa que estabeleça quantos MIPS perfazem um MFLOPS pois tudo depende do tipo do computador. Considera-se em geral que uma operação de ponto flutuante seja executada por 2 a 5 instruções (dependendo da máquina pode chegar a 10).

Uma outra medida surgida ultimamente mais ligada a área de Inteligência Artificial, é o LIPS. Ela mede o número de inferências por segundo (LIPS - Logical Inferences per Second). Novamente a relação entre LIPS e MIPS não é exata e depende do número de instruções que uma dada máquina necessita para perfazer uma inferência, esta relação é da ordem de 1 MLIPS = 100 a 1000 MIPS.

2.3. Classificação dos Supercomputadores quanto a faixa de Custo/Performance

O desempenho de um Supercomputador é medido em termos de MFLOPS. A medida em Mflops reflete sua capacidade em processar resultados em ponto flutuante de 64 ou 32 bits. Uma classificação de computadores é determinada pelo estabelecimento de uma faixa aproximada de desempenho válida para uma dada época de desenvolvimento da tecnologia. Na fig. 5 é apresentado um gráfico [7] baseado numa tabela de Hwang [6] onde aparecem três classes de supercomputadores e uma classe de superminis empregada como referência. As três classes são os SUPERS tais como o Cray, ETA, os QUASE SUPERS tais como os computadores IBM 3020/VF, CDC Cyberplus e os MINISUPERS representados pelas máquinas Alliant e Convex. Segundo a classificação atual um Supercomputador é uma máquina que executa da ordem de 10^1 a 10^3 unidades de Mflops num processamento do tipo científico. É importante salientar que o desempenho de um supercomputador é somente de 5 a 25 por cento da sua performance de pico [6]. Esta baixa performance leva a pesqui-

sas por melhores algoritmos, linguagens, hardware e técnicas de software para a obtenção de melhores desempenhos.

2.4. Medidas de Desempenho

As medidas de desempenho são feitas através de "benchmarks". Somente ultimamente foram desenvolvidos padrões de benchmark para medidas comparativas entre os supercomputadores. Existem o kernel de Livermore (LFK), o LINPACK do Laboratório de Argonne e o SSC dos Laboratórios Sandia entre outros. Para julgar a performance de um sistema o teste deve ser tanto para medir o desempenho do kernel como da aplicação científica. Existem também benchmarks mais gerais tais como o Whetstone e o Dhrystone que são baseados numa coletânea estatística de linguagens, mas que, por serem gerais não expressam corretamente o poder dos supercomputadores, servindo melhor para representar a comparação de ganho nas performances das arquiteturas.

3. ARQUITETURAS DE COMPUTADORES PARA PROCESSAMENTO PARALELO

Concorrência implica em uma das seguintes situações, ou uma combinação destas:

- Eventos paralelos - ocorrem com múltiplos recursos no mesmo intervalo de tempo. São os multiprocessadores que atendem este paralelismo.
- Eventos simultâneos - ocorrem no mesmo instante de tempo. São as arquiteturas matriciais (SIMD).
- Eventos Pipeline - ocorrem em instantes de tempo sobrepostos. O próprio nome indica que são as arquiteturas pipeline.

O Pipeline é um paralelismo temporal de operações sobrepostas. Para obter-se pipeline, é necessário subdividir uma determinada tarefa (ou processo) numa seqüência de sub-tarefas cada uma executada por um estágio de hardware específico que trabalhará concorrentemente com os outros estágios do pipeline. Esta técnica é a mesma empregada numa fábrica onde a linha de montagem trabalha com diversos estágios cujos funcionários executam cada um a mesma tarefa; a medida que o produto, que está sendo fabricado, passa pelos diversos estágios, vai ficando pronto e ao fim do último estágio ele deve estar completamente acabado (Fig. 6). Da mesma forma no Pipeline os dados vão atravessando os estágios (num caso de pipeline aritmético) e sendo processados, enquanto o dado "n" está num estágio, no estágio anterior o dado "n+1", e ao final dos estágios o dado estará pronto, processado. No pipeline de instruções o paralelismo será entre as etapas busca da instrução, decodificação, busca de operando e execução, de instruções diferentes. Exemplos mais conhecidos destas máquinas são os CRAY (CRAY-1, CRAY-2), a série CYBER-200 (Cyber-205) e os computadores VP da Fujitsu (VP-200), VP-400).

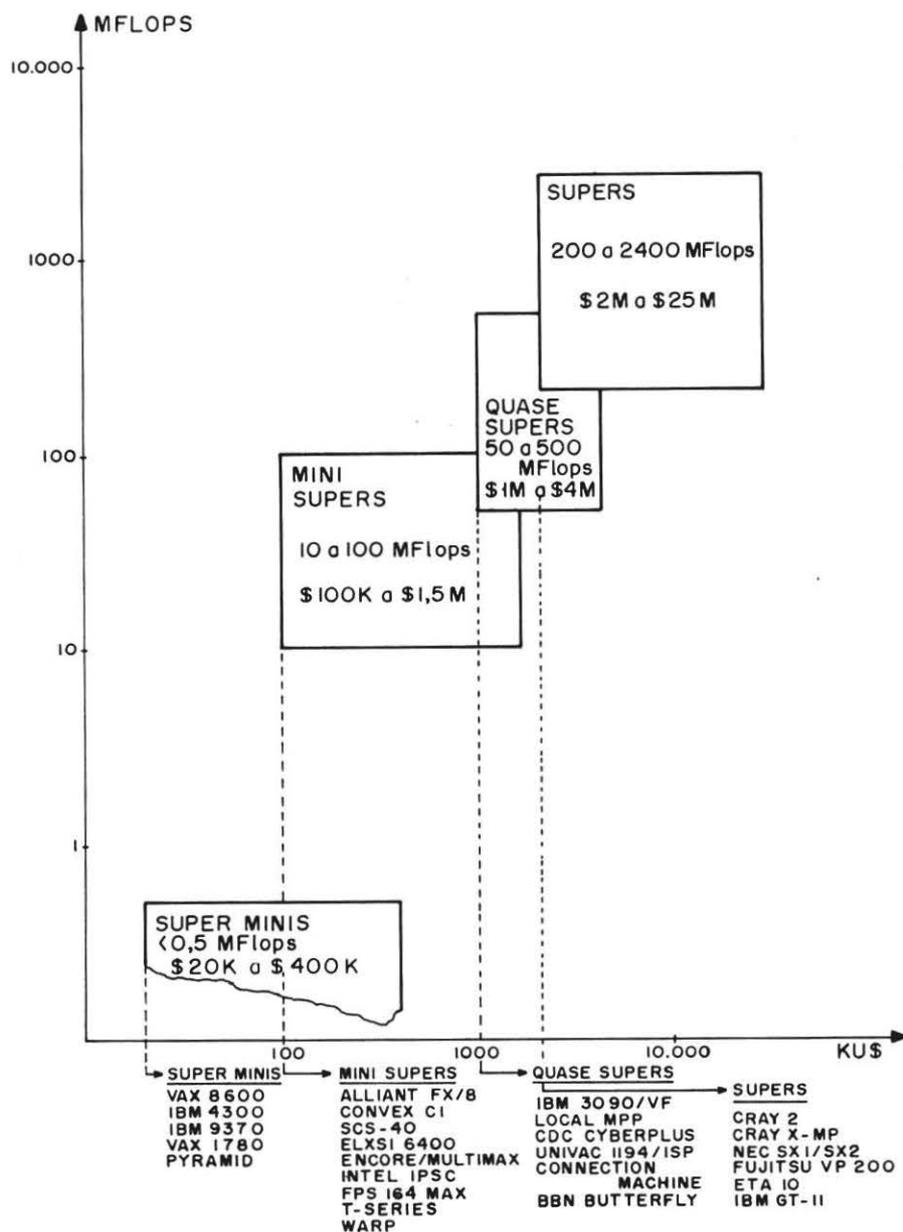


Fig. 5 - Classificação dos Supercomputadores quanto a Custo/Performance

Os Processadores Matriciais (ou Arranjo de Processadores) possuem um paralelismo espacial do tipo síncrono. Esta arquitetura é caracterizada por vários elementos de processamento - EP (ULAs) supervisionadas por uma unidade de controle, de onde emanam as instruções (a mesma instrução) que operam sincronamente sobre os diferentes dados (Fig. 7). Para a comunicação entre os elementos de processamento, com vistas a transmissão de dados e resultados, os processadores matriciais possuem estruturas de interconexão entre os EP permitindo o roteamento dos dados. Normalmente as instruções escalares são executadas diretamente na unidade de controle, enquanto que as instruções vetoriais e matriciais são executadas na matriz de elementos de processamento; estes últimos são circuitos

passivos sem condições de decodificação de instruções, cabendo a unidade de controle o envio dos comandos (instruções) para os EPs executarem.

Os Processadores Matriciais podem ser Associativos cuja diferença básica é o emprego de uma memória associativa com processadores ligados a cada ponto da memória (bit) ou conjunto de posições (bytes, word, string).

Normalmente os computadores SIMD propósitos específicos, são máquinas cujas arquiteturas foram projetadas para obter uma ótima performance em aplicações especiais; e são de difícil programação. Exemplos destas máquinas são o ILLIAC IV, o MPP, o STARAN e o PEPE estes dois últimos associativos. Ultimamente esta

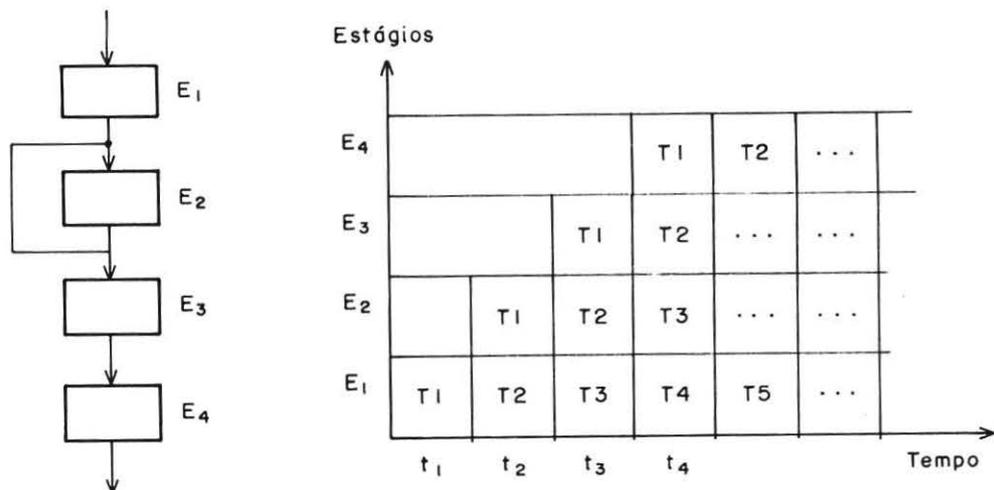


Fig. 6 - Arquitetura Pipeline

arquitetura tem encontrado uma maior aplicação graças ao seu emprego no projeto de arquiteturas para "chips" com vistas, em especial, à área de processamento de imagens.

Os Multiprocessadores, como citado acima, possuem um paralelismo do tipo espacial e assíncrono, isto indica que o paralelismo ocorre devido a que diversos processadores estão processando tarefas concorrentemente porém sem nenhum sincronismo. As tarefas executadas pelos processadores não serão necessariamente as mesmas nem o momento de iniciar este processamento. Por este tipo de arquitetura não ser complexa, não necessitando de tecnologia especial, isto levou a diversos Centros de Pesquisa no Brasil explorarem projetos de máquinas nesta categoria. Pode citar como exemplo o projeto Pegasus do NCE/UFRJ o projeto do LSD da USP, o projeto SUMUS do CPGCC/UFRGS. Do ponto de vista arquitetura os processadores são por exemplo placas com microprocessadores M68000 que interagem entre si através de uma memória global, ou pelo envio de mensagens (Fig.

8). Estes processadores geralmente são idênticos e possuem um Sistema Operacional único que os gerencia. Portanto um multiprocessador é um computador que possui vários processadores que se comunicam e cooperam em diferentes níveis para resolverem uma dada tarefa. Os multiprocessadores dividem-se em "fortemente acoplados", quando a interação entre os processadores é frequente, normalmente este tipo de máquina possui memória global e emprega um cache para cada processador, ou "fracamente acoplados" quando a troca de mensagens entre processadores não é frequente, neste caso, em geral, não é empregada uma memória global. Este último tipo de máquina é denominado por alguns autores de multicomputador [6].

Os principais multiprocessadores conhecidos são o S1 do Laboratório de Laurence Livermore, o Cm* de Carnegie Mellon e os comerciais são o HEP da Denelcor, o Tandem/16, o CRAY X-MP. Além destes existem os que são extensões de uni processadores como o IBM 370/168 MP, o UNIVAC

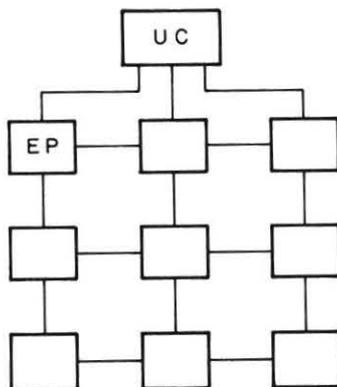


Fig. 7 - Arquitetura Matricial

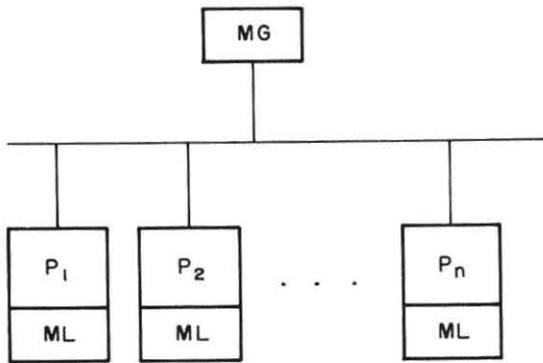


Fig. 8 - Arquitetura Multiprocessador

A importância das máquinas HIPERCÚBICAS vem crescendo rapidamente a ponto de já existirem diversas máquinas comerciais (NCube, IPSC, Butterfly, ...). Mais de cem hipercubos já estão em uso, a maioria em instituições acadêmicas e laboratórios governamentais [10]. Os hipercubos rodam múltiplos programas que atuam sobre múltiplos conjuntos de dados. Internamente a máquina possui unidades de processamento, chamadas de nós, que são independentes e comunicam-se com as outras para executarem programas. Cada nó possui sua própria memória, unidade de ponto flutuante, processador de comunicação, e uma cópia do sistema operacional e do programa de aplicação. Estas máquinas são chamadas de hipercúbicas por possuírem uma organização semelhante a um cubo de "n" dimensões, com um nó em cada vértice (Fig. 9). Não existe ainda uma classificação para esta arquitetura, porém alguns autores [6] a consideram como um multiprocessador fracamente acoplado.

4. FORMAS DE PROCESSAMENTO PARALELO

4.1. Níveis de Paralelismo - Granularidade

O paralelismo no processamento é encontrado em diversos níveis diferentes. Existem diversas classificações com um maior ou menor número de níveis dependendo do autor; Hwang [6] sugere cinco níveis: nível 5 - de processos independentes (Jobs) e programas; nível 4 - sub-processos e pontes de programas; nível 3 - rotinas, sub rotinas e co-rotinas; nível 2 - iterações (loops) e por último, nível 1 - de instruções [6]. O nível mais elevado é o algorítmico e o inferior de instruções é implementado em hardware. Quanto mais baixo o nível mais fina é a granularidade do processamento. O processamento paralelo explora um destes níveis ou uma combinação destes dependendo da máquina. Por exemplo o processamento vetorial emprega o nível 2 de granularidade de itera-

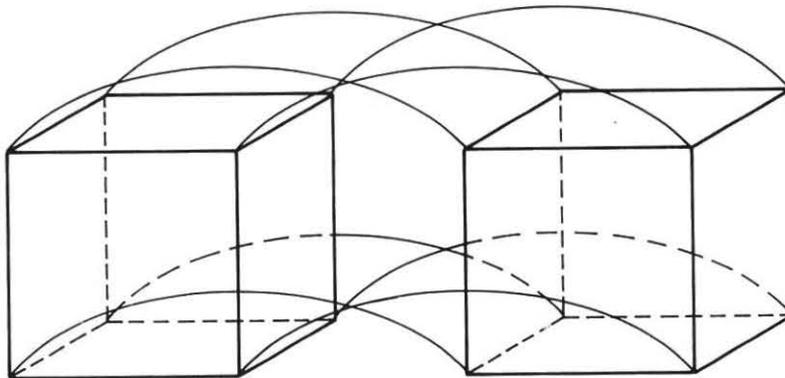


Fig. 9 - Arquitetura Hipercúbica

ções ("loops"). Técnicas de antecipação (look-ahead) em máquinas empregando múltiplas unidades funcionais são do nível 1.

Multiprocessamento é um modo de processamento paralelo que emprega a multiprogramação interativa entre dois ou mais processadores. Multitarefa é um caso especial de multiprocessamento onde é definido um processo de software (uma tarefa) que é um subprocesso ou subprograma portanto do nível 3 ou 4. Máquinas poderosas como o CRAY X-MP possuem paralelismo em todos os níveis, de nível 3, 4 e 5 entre os processadores e de nível 1 e 2 dentro de cada processador. Máquinas do tipo maciçamente paralelas como o MPP possuem somente paralelismo dos níveis inferiores.

4.2. Processamento Pipeline Vetorial

O primeiro exemplo [2] de paralelismo é de um comando DO sendo vetorizado; é um caso típico de uma máquina pipeline vetorial.

Sendo o programa:

```
DO 100 I = 1,N
  A(I) = B(I) + C(I)
  B(I) = 2 * A(I+1)
```

Num processamento escalar estas instruções seriam implementadas sob a forma:

```
INITIALIZE I = 1
10 READ B(I)
  READ C(I)
  ADD B(I) + C(I)
  STORE A(I) ← B(I) + C(I)
  READ A(I + 1)
  MULTIPLY 2 * A(I + 1)
  STORE B(I) ← 2 * A(I + 1)
  INCREMENT I ← I + 1
  IF I < N GOTO 10
STOP
```

Num processamento pipeline vetorial estas operações seriam reduzidas a três operações vetoriais. Há a necessidade de um vetor temporário TEMP (1:N).

```
A(1:N) = B(1:N) + C(1:N)
TEMP(1:N) = A(2:N+1)
B(1:N) = 2 * TEMP (1:N)
```

Esta técnica de substituir um conjunto de instruções seqüenciais por instruções vetoriais é conhecida por VETORIZAÇÃO. Isto pode ser feito por compiladores conhecidos por compiladores vetorizadores, no entanto, é muito complexo um compilador automaticamente detectar todos os pontos passíveis de paralelismo e vetorizar, por isto, estes compiladores possuem em geral uma etapa de vetorização automática e outra interativa onde o usuário auxilia na detecção dos paralelismos.

4.3. Processamento com Processadores Matriciais

A programação deste tipo de máquina é bem mais complexa, e por isto estas máquinas são normalmente empregadas em processamentos específicos, pois as instruções além de levar em con-

ta o próprio código de operação precisam também se ater à conexão (roteamento dos dados), o número de processadores e o conjunto de máscaras.

Descrevemos um exemplo simples [2] para dar uma idéia do funcionamento destas máquinas. Imaginemos a soma $S(k)$ dos K primeiros componentes de um vetor A , variando K de 0 até $n-1$. Seja $A = (A_0, A_1, \dots, A_n)$.

$$S(k) = \sum_{i=0}^k A_i \text{ para } k = 0, 1, \dots, n-1$$

Portanto o objetivo é obter de

$$S(0) = A_0 \text{ até } S(k) = S(k-1) + A_k \text{ para } k = 1, 2, \dots, n-1$$

Assumindo $n = 8$ e que as somas serão feitas num processador matricial com 8 EPs o resultado será obtido em 3 etapas. Inicialmente cada dado A_i está na memória local dos EPi.

Na 1ª etapa A_i é roteado de R_{i+1} e somando a A_{i+1} com o resultado da soma ficando em R_{i+1} .

Na 2ª etapa R é roteado para R_{i+2} (deslocamento de 2) e novamente é somado ficando o resultado em R_{i+2} . Na 3ª e última etapa R é roteado para R_{i+4} (deslocamento de 4) e novamente soma-se com resultado em R_{i+4} . Como resultado final teremos em EP₀ o A_0 , em EP₁ a soma A_0 com A_1 , em EP₂ a soma de A_0, A_1, A_2 , e assim sucessivamente (Fig. 10).

Do ponto de vista de granularidade o processamento com processadores matriciais é semelhante ao com processadores pipeline pois ambos atuam no mesmo nível de paralelismo que é o dos dados (níveis 1 e 2).

4.4. Processamento com Multiprocessadores

Os processamentos anteriores, máquinas pipeline e matriciais, são a nível de dados enquanto que no caso dos multiprocessadores o processamento é a nível de processos (níveis 3, 4 e 5).

Qual é a diferença de um software para multiprocessador, de um para máquina uniprocessadora? Basicamente existem duas diferenças, a arquitetura e o estilo peculiar da programação paralela. Do ponto de vista de arquitetura, o fato de existir um sistema de distribuição (dispatching) dos processos pelos processadores. Quanto ao estilo de programação, ele é peculiar porque prevê a concorrência entre processos que interagem entre si.

Para a divisão de um programa num conjunto de processos que possam ser executados concorrentemente é importante a identificação dos objetos que são partilhados pelos processos. Esta divisão pode ser feita de forma explícita pelo próprio programador, ou de forma implícita quando o compilador detectará o paralelismo.

Uma das formas de estabelecer a concorrência explícita é através de comandos FORK e JOIN. Existem outras formas como "cobegin" "coend" ou "parfor".

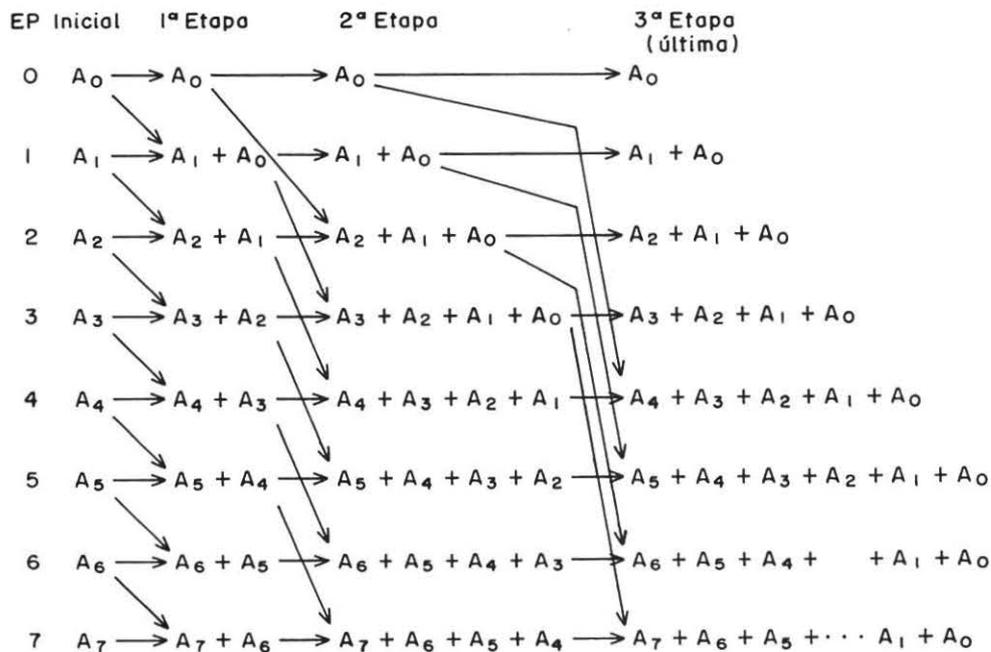
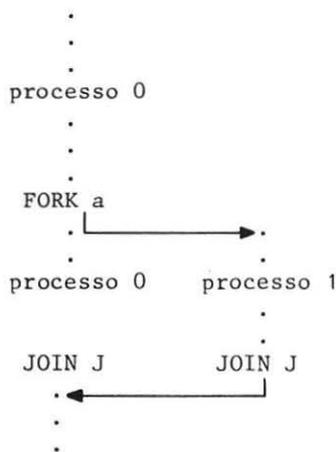


Fig. 10 - Soma de elementos num Processador Matricial



No caso de paralelismo implícito, torna-se necessário reconhecer os processos passíveis de paralelismo pela análise do programa fonte, o que é bem mais complexo. Uma forma de estabelecer isto, é pela análise da "dependência aos dados".

Sincronização é outro aspecto importante nos multiprocessadores; são necessários mecanismos de sincronização eficientes para permitir a interação entre os processos. Esta sincronização será feita em software através de mecanismos, com primitivas do tipo semáforos, implementada diretamente em hardware ou microcódigo.

5. LINGUAGENS PARALELAS

Um dos grandes problemas das máquinas paralelas não é tanto o hardware mas o software, não a-

dianta projetar máquinas paralelas se não existem linguagens paralelas. Por isto, nos últimos anos, um grande esforço está sendo desenvolvido na obtenção de linguagens de alto nível com construções paralelas para facilitar o processamento. Os processadores vetoriais comerciais empregam geralmente extensões de linguagens, como por exemplo o CRAY - 1 com uma extensão do Fortran CFT (Cray Fortran Translator) e o FACOM VP - 200 que usa uma extensão de Fortran - 77.

Algumas linguagens vetoriais foram propostas como a ACTUS de Perrott [2] e o VECTRAN de Paul e Wilson [2]. Infelizmente estas linguagens ainda não foram exaustivamente testadas para estabelecer sua validade ou não. Muitas das linguagens vetoriais/matriciais, além de serem empregadas em processadores pipeline, podem também com adaptações serem empregadas para processadores matriciais que é o caso de Actus e Vectran. Para máquinas matriciais, mais particularmente para o ILLIAC IV, foram desenvolvidas linguagens como o Tranqual (tipo Algol), Glypnir (tipo Algol) e o Illiac-IV Fortran [2].

No caso de linguagens orientadas a processos, portanto para multiprocessadores, só ultimamente surgiram algumas como OCCAM, Modula-2, Pascal Concorrente [6], e outras, porém os supercomputadores comerciais ainda não as aceitam. Aparentemente o grande problema destas linguagens é que os supercomputadores suportam vários tipos (níveis) de paralelismo e as linguagens, até o presente citadas, atendem a um tipo ou a um outro. Devido a esta necessidade

está sendo pesquisado uma linguagem ideal, que seria independente da arquitetura, isto é, poderia ser empregada tanto para máquinas pipeline, matriciais como multiprocessadores. Recentemente foi proposta uma linguagem paralela PAL (Parallel Programming Language) [6] que atenderia estes requisitos.

6. COMPILADORES PARALELOS

A sessão anterior citou os esforços na obtenção de novas linguagens que permitam explorar da melhor forma possível, os tipos de paralelismo que as máquinas apresentam. Como o uso de linguagens levam a necessidade de Compiladores, esta seção abordará a direção da pesquisa nesta área.

As duas formas principais de paralelismo que se empregam atualmente na criação de compiladores inteligentes são: a vetorização e o multiprocessamento [1].

O atual estágio de desenvolvimento dos compiladores vetorizadores já é significativo. O funcionamento é baseado na detecção de laços de DO adaptando-os de forma a possibilitar que a máquina em questão (em geral pipeline ou matricial) execute partes do programa concorrentemente.

A outra forma de paralelismo, bem menos pesquisada, mas que agora está recebendo uma maior ênfase é o multiprocessamento. Este tipo de compiladores busca a detecção de trechos de programas que possam ser executados concorrentemente. Basicamente estes compiladores detectam as dependências de dados nos programas de forma a separar os trechos de programas independentes para que possam ser executados concorrentemente.

Claro que existem outras técnicas que permitem aumentar a concorrência porém estas duas, citadas acima, são as principais. Observa-se que os compiladores paralelos mais modernos exploram ambos tipos de paralelismo. Isto é uma decorrência das próprias máquinas, que como o CRAY X-MP, possui a característica de processamento vetorial pela sua estrutura pipeline mas também de multiprocessamento quando coloca vários processadores pipeline em paralelo.

7. SISTEMAS OPERACIONAIS E CONTROLE DA CONCORRÊNCIA

O gerenciamento dos recursos e o controle da concorrência afetam de forma importante o grau de paralelismo de uma máquina paralela. O gerenciamento dos recursos incluem, o escalonamento (scheduling) dos múltiplos processadores e das unidades funcionais, o acesso e a alocação da memória, e o controle do sistema de E/S. O controle da concorrência trata da execução simultânea de múltiplos processos de software num sistema computacional, [6]. É importante salientar que o controle da concorrência tanto

pode estar a nível de processos como no caso do multiprocessamento, como a nível de vetores para os dados, isto implica em diversas formas diferentes de controle.

A comunicação entre processadores é outro ponto importante para o gerenciamento de supercomputadores. Esta comunicação pode ser feita através de primitivas de sincronização, semáforos de estado, sinais de interrupção, mensagens de tamanho variável, variáveis partilhadas ou "data values". Na verdade os métodos são essencialmente quatro: variáveis partilhadas, troca de mensagens, troca de valores e troca de marcas. Estes métodos são implementados através de memórias partilhadas, registradores partilhados, "buffers" de comunicação, barramentos de interrupção, laços de conexão entre processadores, e outros.

O gerenciamento da E/S e o emprego de um computador frontal para dialogar com o usuário são itens também importantes numa máquina paralela. Existem tarefas de um supercomputador que empregam muito a E/S como no caso de um ambiente de tempo-real; numa situação destas o super não consegue atender esta tarefa eficientemente. O Cray 2, para resolver estes problemas, emprega um processador (foreground processor) encarregado das funções de E/S e de todo o sistema operacional (SO). O Alliant FX/8 emprega processadores interativos (IPs) para a execução de programas interativos, as tarefas de E/S e as atividades de SO.

O balanceamento da carga entre processadores é outro item pesquisado com vistas a otimizar o paralelismo. Este balanceamento pode ser dinâmico quando os processos são alocados aos processadores em tempo de execução, ou então estáticos quando esta alocação for estabelecida por ocasião da compilação.

O acesso e alocação de memórias podem ser executados de diversas maneiras. No computador RP3 a memória global é distribuída entre os processadores, isto é, uma parte da memória local é alocada para formar a memória global. Uma característica interessante desta máquina é que o particionamento desta memória é dinâmico de tal forma que a máquina RP3 pode ser configurada para ser um sistema de memória compartilhada, ou um sistema funcionando por troca de mensagens com memórias locais, ou então um sistema híbrido entre estes dois casos. Quem vai determinar a situação será o ponto de fronteira entre a quantidade de memória global e local do sistema.

O escalonamento dinâmico de processos concorrentes depende das condições de execução; soluções ótimas são impossíveis. Existem várias técnicas heurísticas tais como first-in-first-out (primeiro a chegar primeiro a ser atendido), round robin (atendimento segundo um escalonamento circular), shortest - process - first (processo menor primeiro a ser atendido), least-memory-first (memória menor primeiro atendido), etc. [2]. O escalonamento estático é determinado por ocasião da compilação, porém normalmente os resultados de utilização dos

processadores são pobres). Quanto a mecanismos de controle de atividades algumas técnicas são as mail box (caixas postais), registradores de estado, e caixas (box) de controle de processadores.

8. ALGORITMOS PARALELOS

Um dos itens importantes do paralelismo são os algoritmos paralelos [6]. Na verdade este assunto está relacionado, e muitas vezes é abordado, junto com as aplicações ou formas de programação de supercomputadores. As etapas de definição de um algoritmo podem ser resumidas: primeiro, no modelamento de um problema numa fórmula matemática; segundo, na aproximação numérica e discretização da equação de forma a que possa ser processada por um computador; por último, na partição do algoritmo para que o problema possa ser mapeado numa arquitetura paralela ou vetorial.

Três abordagens diferentes são possíveis no projeto de um algoritmo paralelo. Primeiro a simples conversão de um algoritmo existente numa versão paralela, nem sempre esta solução é adequada, pois a conversão pode gerar um al-

goritmo ineficiente. A segunda situação é o projeto de um novo algoritmo, esta solução permite a obtenção de algoritmos eficientes porém é difícil a obtenção destes. Por último, a abordagem mais prática é uma espécie de combinação dos dois primeiros, inicia com um algoritmo paralelo para um problema semelhante e depois tenta a modificação deste para atender o problema em questão.

O mapeamento de algoritmos para arquiteturas paralelas é em si um assunto tão vasto que é objeto de pesquisa por diversos grupos. Basicamente o problema de mapeamento aparece quando a estrutura de comunicação do algoritmo paralelo difere da estrutura de interconexão da máquina paralela. Um exemplo óbvio desta situação é quando o número de processos que o algoritmo possui for maior que o número de processadores da arquitetura. O grau de paralelismo, a granularidade, a carga devido a comunicação e sincronização, são alguns dos itens que estabelecerão a complexidade na implementação do algoritmo.

O item comunicação, mencionado acima, é outro aspecto importante na definição de um bom algoritmo. Dependendo da situação, caso a comple-

MÁQUINA	SISTEMA OPERACIONAL/LINGUAGEM/COMPILADOR
Cray X-MP	COS/CFT (Fortran) CTSS
Cray 2	Unix-System V/CFT2,C
Cyber 205	Virtual OS/FTN 200
ETA-10	Virtual OS/Fortran, Pascal, C
Fujitsu VP	FACOM VSP/Fortran 77
IBM 3090/VF	XA/VS Fortran V2
Cyberplus	Host NOS2/ANSI 77 Fortran
HEP-1	UNIX III/Fortran 77, C, Pascal
Convex C-1	Unix 4.2 bsd/Fortran 77, C
Alliant/FX	Concentrix/FX Fortran, C, Pascal
Balance 21000	DYNIX/C, Fortran 77, Pascal, Ada
FLEX/32	MMOS/Fortran 77, C, Ada, Concurrent C & Fortran
iPSC-VX	Xenix/C, Fortran, Common Lisp
FPS-T	Host OS/OCCAM
Connection Machine	Host OS/CM-C, CM-Lips
BBN Butterfly	Chrysalis/C, Common Lisp
IBM RP3	UNIX/C, Fortran, Pascal
Culler PSC	UNIX 4.2 csd/Fortran 77, C
NCUBE E/10	AXIS/VERTEX/FORTRAN 77, C

Tabela 1 - Sistema Operacional, Linguagens e Compiladores de algumas máquinas mais conhecidas

xidade de um algoritmo for muito grande, esta torna a implementação deste desinteressante, podendo chegar a situações em que o algoritmo paralelo é mais lento na sua implementação do que sua versão sequencial. Portanto, na definição de um algoritmo é necessário minimizar e balancear entre si, a complexidade da computação, da comunicação, da E/S, e do acesso a memória.

Hwang [6] agrupa os algoritmos em seis famílias principais que são aritmética vetorial/matricial, processamento de sinais ou imagens, processos de otimização, análise estatística, equações diferenciais parciais e funções especiais. Já Kung [11] divide os algoritmos associando com a arquitetura, em sistólicos, SIMD e MIMD sendo que os primeiros possuem granularidade pequena, controle distribuído através de mecanismos locais e são síncronos, os SIMD possuem controle centralizado, síncronos e granularidade de pequena a grande, e por último os MIMD são assíncronos, com granularidade grande e partilham dados.

É importante salientar as pesquisas em algoritmos assíncronos independentes. Estes algoritmos como o de "divisão e conquista" permitem a divisão de um problema até pequenos sub-problemas de tal forma que estes possam ser tratados independentemente ou quase [13].

9. APLICAÇÃO DO PROCESSAMENTO PARALELO

Quais são as necessidades de poder de processamento que levaram ao estudo destas novas máquinas? Na verdade foram as necessidades de executar com o auxílio do computador tarefas mais sofisticadas tais como o estudo sísmico do terreno para localização de petróleo, a determinação do tamanho da bacia de petróleo, a previsão do tempo, e outros itens que abaixo serão enumeradas mais exaustivamente. Uma das áreas mais surpreendentes para novas aplicações destas máquinas é a indústria de entretenimento [8]; por exemplo o emprego destes computadores em conjunto com técnicas gráficas avançadas para a produção de filmes. É tal a necessidade de processamento que pode-se afirmar que "Sem os computadores Superpoderosos, a maioria dos desafios que permitem o avanço da civilização humana não poderão ser alcançados num breve período de tempo" [2].

Lista de Aplicações segundo Hwang [2]:

A. Modelos de Previsão e Simulação

- Prognóstico de Tempo
- Oceanografia e Astrofísica - Exploração de recursos dos oceanos, estudos sobre capacidade de piscosa, estudo da formação da terra, dinâmica das galáxias, etc...
- Sócioeconomia e Governo - Econometria, Engenharia Social, Censo de Governo, Controle do Crime.

B. Projetos de Engenharia e Automação

- Análise de Elementos Finitos - Cálculo de barragens, pontes, navios, aviões, grandes edifícios, veículos espaciais, etc...

- Aerodinâmica - Estudo das turbulências. (A Burroughs e CDC propuseram uma máquina NASF que chegaria a 2 Gigaflops).
- Inteligência Artificial e Automação - Vários são os assuntos pertinentes tais como processamento de imagens, reconhecimento de padrões, visão por computador, reconhecimento de voz, máquinas de inferência, CAD, CAM, CAI, OA (Projeto Automatizado por Computador, Manufatura Automatizada por Computador, Instrução Automatizada por Computador, Automação de Escritório), robótica inteligente, sistemas especialistas, engenharia do conhecimento.

Representativo deste esforço é o projeto japonês para desenvolvimento do Computador de 5ª geração.

- Aplicações em Sensoriamento Remoto - Análise de imagens de satélite, para obtenção de informações sobre a agricultura, florestas, geologia, fontes hídricas, etc... A NASA encomendou o MPP (Massively Parallel Processor) que é uma máquina que possui 16k processadores com palavra de 1 bit para processamento de imagens de satélite.

C. Exploração de Fontes de Energia

- Exploração Sísmica - Empregado especialmente pelas Companhias Petrolíferas para determinação do local de poços de petróleo.
- Modelagem de Reservatórios - Determinação do volume de uma bacia de petróleo (por exemplo).
- Potência da Fusão de Plasma - A fusão nuclear necessita de máquinas poderosas para modelar a dinâmica do plasma.
- Segurança de Reatores Nucleares - Análise das condições do reator, Controle automático, Treinamento através da simulação de operações, Atuação rápida em casos de acidentes.

D. Pesquisas Médicas, Militares e Básicas

- Tomografia Assistida por Computador - Modelamento do corpo humano via tomografia para permitir seu estudo tridimensional determinando anormalidades.
- Engenharia Genética - Foi construído o Cyto-computer que permite pesquisar mutações genéticas.
- Pesquisa de Armas e Defesa - Projeto de Armas e Defesa - Projeto de armas nucleares multi-cabeças, simulação dos efeitos das armas em especial as radioativas, processamento de sinais de radares para comando de mísseis antibalísticos, geração automática de mapas, acompanhamento de submarinos submersos, etc
- Problemas de Pesquisa Básica - Em Química, Física e Engenharia tais como mecânica quântica, mecânica, estatística, química de polímeros, crescimento de cristais, análise de trajetória de partículas, dinâmica de fluidos, teoria do campo quântico, dinâmica molecular, equações de circuitos de grande escala, distribuição de conexões em circuitos VLSI, etc...

No Brasil já se encontram diversas necessidades desde processamento paralelo tais como as da Petrobrás para determinação de poços pela con-

figuração do terreno, cálculo de bacias de petróleo e cálculo das estruturas das plataformas de perfuração em alto-mar; dos Ministérios da Agricultura, Marinha e Aeronáutica na previsão de tempo; das Centrais Elétricas quanto à distribuição da energia, entre outros.

10. CONCLUSÕES E EVOLUÇÃO

O presente levantamento pretendeu apresentar a área de Processamento Paralelo, discutindo seus diversos tipos de arquiteturas, processamento, linguagens, compiladores, controle, algoritmos, apresentando seus problemas e evoluções. É óbvio, no entanto, que, dada a abrangência do assunto, não era possível abordar to

dos itens no espaço deste levantamento. No entanto, espera-se que através desta rápida abordagem tenha sido esclarecido alguns tópicos e principalmente chamado a atenção para a importância do assunto. É certo que o futuro da informática passará pelo processamento paralelo, e portanto o Brasil precisa preparar-se para esta nova evolução da informática.

Não foram apresentadas as redes neurais, com o modelo coneccionista de arquitetura, nem as máquinas de fluxo de dados e de redução entre outros. Estas modalidades de paralelismo estão sendo objeto de pesquisas porém sua implementação comercial ainda é incipiente ou não existe. As arquiteturas apresentadas possuem como característica principal o fato de existi

Modelo	Arquitetura	Nº de Processadores	Performance de Pico (flops)
Cray X-MP/4	Multiprocessador c/Mem. Partilhada	4	840 M
Cray 2	Multiprocessador c/Mem. Partilhada	4 + 1E/S	2G
Cray 3	Multiprocessador c/Mem. Partilhada	16	16G
Cyber 205	Uniprocessador c/4 pipelines Vetoriais	1	400 M
ETA 10	Multiprocessador c/Mem. Partilhada	8 + 18E/S	10G
VP200	Uniprocessador c/múltiplos pipeline	1	533 M
IBM 3090/400/VF	Multiprocessador c/Mem. partilhada	4	480 M
BBN Butterfly	Multiprocessador c/rede butterfly	256	256 Mips
Connection Machine	SIMD hipercubo	64 K	250 M
Alliant FX/8	Multiprocessador c/Mem. Partilhada	8 + 12 (interação)	94 M
Ballance 21000	Multicomputador	30	2 M
Convex	Uniprocessador c/hardware vetorial	4 + 5E/S	20 M
N Cube/10	Multicomputador mem. distribuída hipercubo	1024	500 M
Culler-PSC	Multiprocessador c/Mem. partilhada	2	5 M
Warp	array sistólico	10	100 M

Tabela 2 - Características de algumas máquinas paralelas

tirem máquinas sendo fabricadas e comercializadas.

No campo da tecnologia grandes esforços de pesquisas estão sendo feitos no desenvolvimento de circuitos GaAs, interconexões óticas e redes neurais; a próxima geração de computadores Cray, o Cray 3, será fabricado com circuitos GaAs.

O maior desafio, no entanto, é o desenvolvimento de linguagens, compiladores, sistemas operacionais, em última análise, software paralelo. Só assim será possível o uso destas máquinas com desempenho. Há a necessidade de criar uma cultura em Processamento Paralelo, alguns autores sugerem inclusive a criação de uma ciência [6].

O gráfico da figura 11 apresenta uma visão da provável evolução dos computadores paralelos. Este gráfico baseado em dados do fabricante Apollo mostra claramente através das linhas de equipotência a evolução esperada das futuras máquinas. Observa-se, por exemplo, que o poder de processamento do IBM 360 da década de 60, já estava contido numa pastilha do microprocessador M68.000 em 1978. Da mesma forma hoje um minisuper possui a mesma capacidade de um CRAY-1 de 1978 e pelo ano de 1993 terão as estações de trabalho. Isto permite prever que provavelmente na década 2.000 teremos um CRAY-1 pelo preço de um microcomputador pessoal dos dias de hoje...

No Brasil nos últimos meses centros de pesquisa, indústrias, usuários e entidades do governo tem se reunido para discutir o assunto e traçar metas para o desenvolvimento da área no Brasil. Em julho de 86 a SBC e a FINEP pro-

moveram uma primeira reunião para análise do assunto que foi logo seguida em outubro por uma reunião mais abrangente no CTI - Campinas, onde foram discutidos os aspectos mais políticos e estratégicos. Em maio de 87 ocorreu o 1º Simpósio de Processamento Paralelo em Gramado-RS, de cunho técnico-científico, onde mais de 40 trabalhos foram apresentados. Resultado destas reuniões o CNPq organizou uma comissão para elaboração do documento "Programa Giga" que é uma proposta estratégica em processamento de alto desempenho. Posteriormente, a SEI criou uma comissão, nº 30 de "Processadores", onde uma subcomissão tratou da área de Arquiteturas Não-Convencionais cujo documento será empregado como contribuição na formulação do próximo PLANIN. Todos estes fatos acima enumerados demonstram que já existe uma preocupação pela área no Brasil, porém as metas sugeridas nos documentos não serão alcançadas sem um efetivo apoio dos órgãos financiadores aos projetos da área.

OBS.: Partes do presente levantamento foram extraídas de um artigo apresentado no Congresso da SUCESU 88 com o título de "Análise do Processamento Paralelo".

Agradeço ao Alexandre Carissimi e João Cesar Netto pelas sugestões e discussões sobre a forma e assuntos do presente levantamento.

BIBLIOGRAFIA

Tendo em vista a impossibilidade de citar as inúmeras referências neste levantamento, foram apontadas algumas, como [2] e [6], que possuem nas suas referências estas informa-

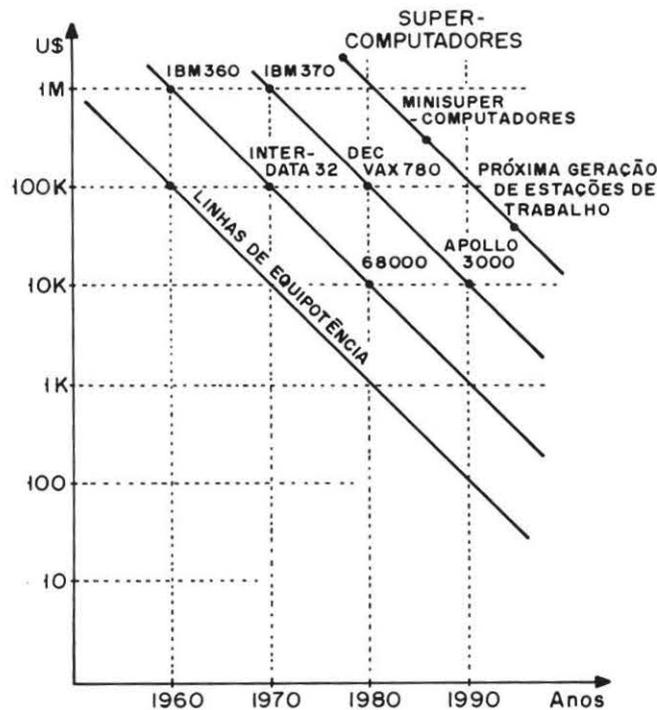


Fig. 11 - Evolução de Computadores: Gráfico Variação do Custo/Anos, Curvas de Potência de Processamento

ções.

- [1] Padua, D.A. & Wolfe, M.J., Advanced Compiler Optimizations for Supercomputers, Communications of the ACM, v. 29, n. 12, dec. 86, p. 1184-1201.
- [2] Hwang, K. & Briggs, F.A., Computer Architecture and Parallel Processing, McGraw-Hill, 1984.
- [3] Verity, J.W., A New Slant on Parallel Processing, Datamation, Fev. 15, 1987, p. 79-84.
- [4] Flynn, M.J., Some Computer Organization and their Effectiveness, IEEE Trans. on Computers, set. 72, p. 948-960.
- [5] Navaux, P.O.A., Introdução as Arquiteturas de Computadores para Processamento Paralelo, VI Jornadas de Atualização em Informática, VIII Congresso da SBC, Julho 87.
- [6] Hwang, K., Advanced Parallel Processing with Supercomputer Architectures, Proceeding of the IEEE, Vol, 75, n. 10, out. 87, p. 1348-1379.
- [7] Catto, J.C., Processadores Paralelos, Documento interno apresentado em 19.01.88 reunião da Comissão nº 30 Processadores da SEI.
- [8] Paul, G., The Special Issue on Supercomputers - Their Impact on Science and Technology, Proceeding of the IEEE, vol. 72, n. 1, jan 84, p. 4-5.
- [9] Dewitt, P., Fast and Smart, Time, 28 março 88, p. 33-38.
- [10] Wiley, P., A Parallel Architecture comes of age at last, IEEE Spectrum, jun. 87, p. 46-50.
- [11] Kung, H.T., The structure of parallel algorithms, Advances in Computers, v. 18, New York, Academic Press, 1980, p. 65-112.
- [12] Kogge, P.M., The Architecture of Pipelined Computers, McGraw-Hill, 1981.
- [13] Voigt, R.G., Where are the parallel algorithms? AFIPS-NCC 85, p. 329-333.

"O Coordenador do II SBAC-PP agradece ao Professor Philippe O. A. Navaux a elaboração deste levantamento, que a ele foi solicitado, com o propósito de prover um caráter técnico introdutório ao documento representado pelos Anais do II Simpósio Brasileiro de Arquitetura de Computadores - Processamento Paralelo. O Coordenador deste evento deseja também agradecer à Financiadora de Estudos e Projetos - FINEP, na pessoa do Professor Helio Otávio Pinto Guedes, pelas construtivas sugestões dadas neste sentido."