

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Taxonomias de Heterogeneidade

Kalinka Regina Lucas Jaquie Castelo Branco
Marcos José Santana
Regina Helena Carlucci Santana
Sarita Mazzini Bruschi

Nº 257

RELATÓRIOS TÉCNICOS



São Carlos – SP
Mai./2005

SYSNO	1455882
DATA	/ /
ICMC - SBAB	

Taxonomias de Heterogeneidade

Kalinka Regina Lucas Jaquie Castelo Branco

Marcos José Santana

Regina Helena Carlucci Santana

Sarita Mazzini Bruschi

{kalinka, mjs, rcs, sarita}@icmc.usp.br

Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação e Estatística

Laboratório de Sistemas Distribuídos e Programação Concorrente

C.P. 668, CEP 13560-970, São Carlos – SP, Brasil

Resumo

Este relatório apresenta e discute criticamente as diferentes taxonomias de heterogeneidade presentes na literatura, visando com isso propor uma nova taxonomia, mais compacta e abrangente. Existem na literatura diversas taxonomias, entretanto, a maioria delas não se adequa quando se leva em consideração o grau de heterogeneidade presente em um sistema computacional distribuído. É mostrado ainda neste relatório que as taxonomias existentes não englobam todos os conceitos necessários para a correta classificação de um sistema, e quando isso ocorre essa taxonomia se torna tão complexa e confusa que inviabiliza a classificação do sistema. Uma nova taxonomia é então proposta de modo a tornar a classificação mais concisa e coerente fazendo uso não somente das taxonomias previamente apresentadas na literatura, mas também de novos conceitos como o grau de heterogeneidade.

Maio / 2005

Sumário

1.	Introdução.....	1
2.	Taxonomias de Heterogeneidade	1
3.	Proposta de uma Nova Taxonomia.....	9
4.	Considerações Finais	13
	Referências Bibliográficas.....	14

1. Introdução

Computação heterogênea é definida como uma forma especial de computação paralela distribuída, computação essa que executa utilizando um número de computadores autônomos diferentes (inclusive em modo de execução) e interconectados.

Os diversos tipos de aplicações geralmente requerem diferentes tipos de computação. Da mesma forma diferentes sub-tarefas de uma aplicação podem ter diferentes requisitos computacionais que resultam em diferentes necessidades de capacidades de máquinas. Isso faz com que o uso de sistemas computacionais distribuídos heterogêneos seja cada vez mais necessário quando se quer obter um melhor desempenho.

Com o surgimento desses sistemas computacionais distribuídos heterogêneos, aparecem também as diferentes composições desses sistemas, tanto em nível configuracional quanto em nível arquitetural. Com o intuito de organizar essas composições: modo de execução, modelo de máquina, características de aplicações, mapeamento das aplicações e o grau de heterogeneidade, surgem as taxonomias de heterogeneidade.

Essas taxonomias derivam e baseiam-se não somente em classificações de arquitetura como as de Flynn (Flynn, 1996) e Duncan (Duncan, 1990), mas também em classificações de escalonamento como a de Casavant e Kuhl (Casavant & Kuhl, 1988) entre outros e são detalhadas na próxima seção. Na seção 3 defini uma nova taxionomia para a classificação de ambientes heterogêneos e a seção 4 apresenta as considerações finais.

2. Taxonomias de Heterogeneidade

Várias taxonomias existentes na literatura buscam organizar as propostas apresentadas para a computação heterogênea. Entretanto, em todas as taxonomias apresentadas, a única heterogeneidade que é levada em consideração é a heterogeneidade arquitetural. Exemplificando essas taxonomias têm-se:

1. (Watson et al, 1996) propuseram uma classificação que divide os sistemas em: sistemas computacionais heterogêneos *mixed-machine* e sistemas computacionais heterogêneos *mixed-mode*. Sistemas *mixed-machine* são conjuntos de máquinas heterogêneas independentes, interligadas por uma rede

de alta velocidade, a fim de funcionar como um metacomputador. Por outro lado, sistemas *mixed-mode* referem-se a um sistema de processamento paralelo simples, cujos processadores são capazes de executar em modos de paralelismo síncrono SIMD (Single Instructions Multiple Data) ou assíncrono MIMD (Multiple Instructions Multiple Data), podendo até mesmo chavear entre esses dois modos em nível de instrução. Desse modo, sistemas computacionais *mixed-machine* denotam heterogeneidade espacial enquanto que os sistemas computacionais *mixed-mode* denotam heterogeneidade temporal.

Embora essa classificação seja concisa e simples, efetuando duas diferenciações de sistemas (quanto à organização principalmente), as duas abordagens simplesmente diferenciam máquinas puramente paralelas de sistemas de máquinas sem processamento intrinsecamente paralelo, não efetuando uma diferenciação quanto a presença ou não de conjuntos mais ou menos heterogêneos.

2. (Ekemecic, Tartaja & Milutinovic, 1996) por sua vez propuseram uma classificação que visa estender a classificação de Watson. Essa classificação é chamada de EM³ (EMMM= execution mode, machine model) e classifica os sistemas computacionais em duas direções ortogonais: modo de execução e modelo da máquina.

O modo de execução (*execution-mode*) é caracterizado pelo tipo de paralelismo encontrado na arquitetura (por exemplo, MIMD ou SIMD). A heterogeneidade estruturada nesse modo pode ser organizada tanto em espacial, quando os modos de execução são diferenciados por unidades de hardware, quanto em temporal, quando alterna os modos de execução através de instruções de máquina.

O modelo da máquina (*machine-model*) define as diferenças entre a arquitetura e o desempenho da máquina. Por exemplo, uma estação de trabalho e um computador pessoal possuem diferentes arquiteturas. Além disso, duas máquinas com a mesma arquitetura, mas com relógios (*clocks*) e/ou quantidades de memória principal diferentes produzem desempenhos desiguais e também são considerados com diferentes modelos de máquina. A heterogeneidade baseada em modelos de máquina é naturalmente espacial.

A Figura 1 apresenta o esquema gráfico dessa taxonomia.

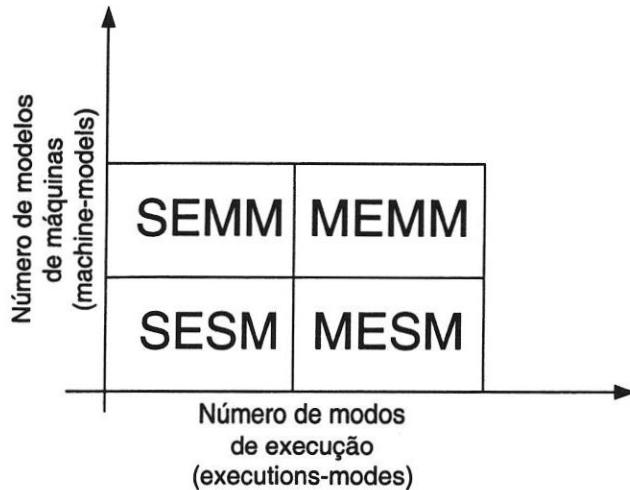


Figura 1 - Taxonomia EMMM e suas subdivisões

Os sistemas computacionais heterogêneos são enquadrados dentro dessa taxonomia contando-se o número de modos de execução (Execution-Modes – EM) e o número de modelos de máquinas (Machine-Models – MM). Desse modo, são quatro as subdivisões da taxonomia EM³:

- SESM (Single Execution Single Mode)
- SEMM (Single Execution Multiple Mode)
- MESM (Multiple Execution Single Mode)
- MEMM (Multiple Execution Multiple Mode)

Dentro da subdivisão SESM enquadram-se bem os sistemas totalmente homogêneos, enquanto que na SEMM podem-se observar os sistemas compostos de diferentes arquiteturas (ou diferentes clocks) com o mesmo modo de execução.

Ambas as subdivisões SEMM e MEMM são compostos de modelos de máquinas distintos, entretanto somente a subdivisão MEMM inclui diferentes modelos de máquinas e diferentes modos de execução.

A subdivisão MESM é a que corresponde aos sistemas com múltiplos modos de execução, entretanto mesmo modelos de máquinas, abordando assim somente a heterogeneidade temporal, uma vez que sistemas compostos de diferentes arquiteturas enquadram-se dentro da subdivisão MEMM.

A classificação EM³ divide-se em quatro classes e também leva em consideração o modo de execução e o tipo de arquitetura do sistema computacional em questão. Entretanto, essa taxonomia não leva em

consideração as características das aplicações que serão executadas bem como a forma com que essas aplicações serão mapeadas nos processadores.

3. (Eshaghian, 1996) propõem uma taxonomia que agrupa os sistemas computacionais em:

- Sistemas Computacionais Heterogêneos (SCH)
- Redes Computacionais Heterogêneas (RCH)

A classe (SCH) subdivide-se em: sistemas computacionais heterogêneos *multimodes* e sistemas computacionais heterogêneos *mixed-modes*. Os sistemas *multimodes* podem executar simultaneamente em modo SIMD ou MIMD e exibem uma heterogeneidade espacial em uma única máquina. Já os sistemas *mixed-mode* efetuam o chaveamento entre os modos SIMD e MIMD, exibindo dessa maneira uma heterogeneidade temporal em uma única máquina.

Da mesma forma, a classe (RCH) subdivide-se em redes computacionais heterogêneas *multimachine* e redes computacionais heterogêneas *mixed-machine*. As redes *multimachine* caracterizam-se por sistemas computacionais distribuídos homogêneos, enquanto que as redes *mixed-machine* são compostas por sistemas computacionais distribuídos heterogêneos.

Essa classificação divide os sistemas somente pelas características das máquinas (se são homogêneas ou heterogêneas) e pelo tipo de processamento que essas máquinas executam. Apesar de efetuar essa divisão, existem redes computacionais que podem ser homogêneas em termos arquiteturais e configuracionais. Mais uma vez não existe a preocupação com as características das aplicações nem com a forma de mapeamento dessas aplicações ao conjunto de processadores existentes, deixando a classificação, em termos, incompleta.

4. (Kafil & Ahmad, 1997) definiram uma outra taxonomia que leva em conta os diversos tipos de algoritmos que buscam o mapeamento de tarefas em processadores. Essa taxonomia classifica hierarquicamente, em um primeiro nível, os algoritmos em ótimos e sub-ótimos. Os algoritmos ótimos, por sua vez, podem ser ainda classificados como restritivos ou não restritivos. Algoritmos

restritivos proporcionam ótimas soluções em tempo polinomial¹ restringindo-se a estrutura do programa, ao processador ou a ambos. Soluções não restritivas, por outro lado, consideram o problema de modo geral e proporcionam soluções ótimas, mas não em tempo polinomial.

Os algoritmos sub-ótimos podem ser classificados como aproximativos ou heurísticos. Os aproximativos utilizam o mesmo modelo do algoritmo ótimo, mas almejam boas soluções ao invés de procurar por todo o conjunto de soluções a solução ótima. As aproximações heurísticas utilizam parâmetros especiais que afetam indiretamente o sistema. A Figura 2 apresenta de forma gráfica a taxonomia em questão.

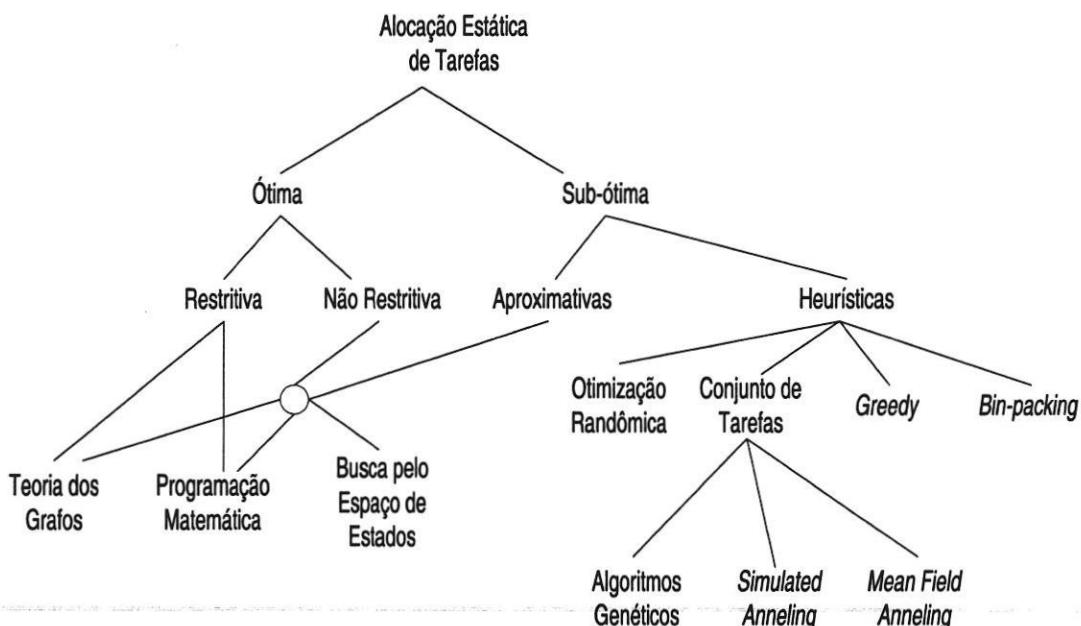


Figura 2 - Uma classificação para algoritmos de alocação de tarefas

Em contraposição às demais classificações apresentadas anteriormente, essa taxonomia aborda basicamente as estratégias de mapeamento, deixando de lado a arquitetura e características intrínsecas às aplicações, que devem ser levadas em consideração juntamente com as estratégias de mapeamento.

5. (Braun et al., 1998) propuseram uma nova taxonomia que visava contemplar não somente as diferenças arquiteturais das máquinas, mas também as heurísticas

¹ Tempo polinomial: De acordo com Ibaraki (1988), comumente um problema é dito ser fácil se existe um algoritmo com complexidade de tempo $O(N^k)$ para uma constante k , onde N é o tamanho do problema.

de mapeamento de subtarefas às máquinas heterogêneas. A essa nova taxonomia foi dado o nome de Purdue Heterogeneous Computing Taxonomy.

A taxonomia de Purdue, assim como as anteriormente mencionadas, constitui uma extensão e modificação da primeira taxonomia apresentada. Essa taxonomia subdivide-se em três categorias:

- 1^a- modelo de caracterização das aplicações;
- 2^a - modelo de caracterização da plataforma;
- 3^a - caracterização da estratégia de mapeamento.

A taxonomia de Purdue utiliza uma organização plana, distinguindo-se das demais que apresentam uma organização hierárquica e, sendo assim, para a caracterização das aplicações definem-se as seguintes características:

- Tamanho da aplicação;
- Tipo da aplicação;
- Padrões de comunicação;
- Disponibilidade de dados;
- Finalizações;
- Modelo de tempo de execução;
- Heterogeneidade das meta-tarefas;
- Múltiplas versões;
- Prioridades;
- Requisitos de qualidade de serviço;
- Heterogeneidade das sub-tarefas;
- *Profile* da tarefa;
- Distribuição temporal.

Seguindo a mesma linha, a caracterização da plataforma dá-se também por uma taxonomia plana, e as características são dadas por:

- Benchmarks analíticos;
- Tempo de comunicação;

- Send/receive concorrentes;
- Rede de interconexão;
- Arquitetura da máquina;
- Heterogeneidade da máquina;
- Número de conexão;
- Número de máquina;
- Sobreposição de comunicação/computação;
- Controle do sistema;
- Compatibilidade de tarefas.

A terceira categoria define as características utilizadas para descrever as estratégias de mapeamento:

- Suporte ao modelo de aplicação;
- Tempo de comunicação;
- Controle de localização;
- Precocidade de acesso;
- Dependências;
- Duplicação;
- Estático/dinâmico;
- Localização de execução;
- Tempo de execução;
- Tolerância a falhas;
- Retorno;
- Função objetivo;
- Suporte ao modelo de plataforma;
- Preempção;
- Re-mapeamento.

Em contrapartida às demais taxonomias que são consideradas hierárquicas, a taxonomia de Pardue apresenta-se de forma plana. O objetivo dessa taxonomia é sanar as falhas apresentadas pelas taxonomias anteriores, abordando características tanto das aplicações quanto da plataforma e das estratégias de mapeamento. Entretanto, apesar de mostrar-se mais completa que as taxonomias anteriores, essa taxonomia não é tão clara, nem mesmo concisa, e deixa a desejar no cerne da diferenciação entre sistemas homogêneos e heterogêneos.

Sendo assim, as taxonomias apresentadas anteriormente são as encontradas na literatura aberta e buscam classificar e organizar adequadamente os diversos sistemas existentes. Maiores detalhes podem ser encontrados em (Ekemecic, Tartaja & Milutinovic, 1996) (Kafil & Ahmad, 1997) (Braun et al., 1998) (Khokhar et. al., 1993) (Ambrosious et. al., 1996) (El-Rewini, 1995) (Freund & Conwell, 1990).

De modo geral, o que se apresenta na literatura são taxonomias que quando concisas e de fácil entendimento não contemplam todas as características presentes no contexto da computação heterogênea, e quando mais completas tornam-se demasiadamente complexas dificultando a organização das novas idéias.

Em todas as classificações apresentadas a única heterogeneidade que é levada em consideração é a arquitetural. Entretanto, estudos realizados e apresentados demonstram que além da heterogeneidade arquitetural, um sistema computacional distribuído heterogêneo pode apresentar ainda uma heterogeneidade configuracional, e principalmente no contexto deste trabalho uma heterogeneidade temporal ou dinâmica.

Por heterogeneidade temporal ou dinâmica pode-se compreender que em determinado instante ou momentos particulares máquinas arquiteturalmente e configuracionalmente homogêneas tornam-se máquinas temporalmente ou dinamicamente heterogêneas. Assim, um sistema computacional pode *estar* ou *não estar* heterogêneo (Branco et. al., 2003a) (Branco et al., 2003b). Com a visão dada pela heterogeneidade temporal passa-se a levar em conta não somente a *compatibilidade* das máquinas, mas também a *capacidade* das mesmas.

Partindo dessas premissas, uma nova taxonomia é proposta adequando as taxonomias existentes para que possam refletir, a partir de então, o grau de

heterogeneidade apresentado por um sistema levando inclusive em consideração se o sistema está ou não heterogêneo.

3. Proposta de uma Nova Taxonomia

Tomando como base o grau de heterogeneidade e a inexistência de uma taxonomia que leve em conta esse grau, propõe-se uma nova taxonomia que envolva não só as características mencionadas nas taxonomias já existentes bem como a heterogeneidade de um sistema, não deixando, entretanto, que a taxonomia torne-se demasiadamente complexa.

A taxonomia aqui proposta busca utilizar-se dos estudos anteriormente apresentados na literatura (Ekemecic, Tartaja & Milutinovic, 1996) (Kafil & Ahmad, 1997) (Braun et al., 1998) (Khokhar et. al., 1993) (Ambrosious et. al., 1996) (El-Rewini, 1995) (Freund & Conwell, 1990) (Flynn, 1996) (Duncan, 1990) (Casavant & Kuhl, 1988) como base e, estender seus conceitos para que os diversos conjuntos de sistemas heterogêneos possam ser mensurados e enquadrados de acordo com a homogeneidade/heterogeneidade que venham a apresentar.

Dessa maneira, os sistemas computacionais podem estar divididos em dois grandes grupos: Monousuário e Multiusuários. Sistemas monousuários constituem sistemas onde existe apenas um único computador que pode executar uma ou mais tarefas computacionais.

De modo similar ao apresentado na taxonomia proposta em (Watson et al, 1996), a máquina que compõem esse sistema monousuário pode ser considerada *mixed-mode* ou *multimode*. *Mixed-mode* referem-se a um sistema de processamento paralelo simples, cujos processadores são capazes de executar em modos de paralelismo síncrono SIMD (Single Instructions Multiple Data) ou assíncrono MIMD (Multiple Instructions Multiple Data), enquanto que por *Multimode* esses modos de paralelismo podem ser executados simultaneamente. Esses sistemas são sempre considerados homogêneos, uma vez que são compostos por apenas um elemento de processamento.

Multiusuários, por sua vez, compreendem os que permitem a presença de diversos usuários conectados ao sistema e sistemas compostos por mais de um elemento de processamento. Nesse conjunto enquadram-se tanto as máquinas intrinsecamente paralelas quanto os sistemas compostos por diversas máquinas interligadas por uma rede de comunicação.

Quando as máquinas que compõem o sistema são simplesmente interconectadas por uma rede de comunicação elas compõem uma rede de computadores ou se possuírem um sistema operacional distribuído compõem um sistema computacional distribuído.

Como os sistemas distribuídos podem ser heterogêneos, caso os sistemas operacionais também sejam distintos, uma máquina pode interagir de maneira eficiente com a outra máquina (como se fosse apenas uma) através de ferramentas como o PVM (Parallel Virtual Machine) ou o MPI (Message Passing Interface).

Em ambientes de passagem de mensagens, a comunicação se dá através do envio e recebimento de mensagens através da rede, seguindo regras do protocolo de comunicação entre vários processadores que possuem memória própria. Juntas, as primitivas *send/receive* executam ações de passagem de mensagens entre um par de processos (Coulouris et al., 1994). Nesses ambientes, o programador é responsável pela sincronização dos processos.

A programação em ambientes de passagem de mensagens se dá através do uso de bibliotecas de comunicações que são extensões das linguagens convencionais como o C ou Fortran. Inicialmente essas bibliotecas foram desenvolvidas para uso com processamento maciçamente paralelo (MPP) como uma tentativa de se padronizar um ambiente para possibilitar a portabilidade entre os sistemas de vários fabricantes. A idéia era definir um conjunto de funções independente da máquina que está sendo utilizada e implementá-las em várias plataformas de *hardware* (Geist et al., 1994; Santana et al., 1997).

Ao contrário do MPP, os sistemas distribuídos fornecem um ambiente heterogêneo onde podem variar a arquitetura; o formato de dados; a potência computacional; a carga de trabalho em cada máquina e a carga de trabalho nos meios de comunicação.

Ainda presente dentro deste grupo multiusuário, encontram-se as máquinas maciçamente paralelas.

De modo análogo ao apresentado na taxonomia proposta por (Ekemecic, Tartaja & Milutinovic, 1996), os sistemas multiusuários podem ser *Single Machine* ou *Mixed-Machine* de acordo com as características apresentadas pelas máquinas que irão compor esses sistemas ou pela diversidade dos elementos de processamento que apresentam.

Em *Single Machine* enquadram-se os sistemas compostos por um único modelo de máquina, enquanto que os sistemas compostos por diferentes modelos de máquinas enquadram-se em *Mixed-Machine* (incluindo-se aqui tanto diferenças arquiteturais quanto diferenças de *clock*).

Após sua classificação em termos de especificidade de máquina, esses podem ainda ser classificados quanto ao tipo de execução apresentada *Mixed-Mode* ou *Multi-Mode*.

Sistemas que são classificados como *Single Machine* e *Mixed-Mode* podem ser considerados homogêneos em termos arquiteturais, ao passo que os classificados como *Single Machine* e *Multi-Mode* devem ser classificados como heterogêneos, também em termos arquiteturais.

De modo análogo os sistemas classificados como *Mixed-Machine* podem também ser considerados *Mixed-Mode* ou *Multi-Mode*. Uma vez que neste subgrupo as máquinas distinguem-se no mínimo arquiteturalmente, esses sistemas são considerados heterogêneos em termos arquiteturais.

Entretanto, como mencionado anteriormente, em relação ao tempo esses sistemas podem ser classificados como: homogêneo, virtualmente homogêneo ou heterogêneo como apresentado na Figura 3.

Essa heterogeneidade, expressada pelo grau de heterogeneidade (Branco et. al., 2003a), indica a diversidade entre as máquinas que compõem o sistema, mas não somente em termos de arquitetura, mas sim em termos de capacidade de recursos, uma vez que essas máquinas distam umas das outras em termos não somente de quantidade, quanto em características arquiteturais, permitindo assim que os sistemas possam ser enquadrados em um dos três níveis de heterogeneidade.

Uma vez definida a heterogeneidade do sistema computacional, pode-se então definir com uma maior flexibilidade o tipo de escalonamento e os parâmetros que devem ser utilizados.

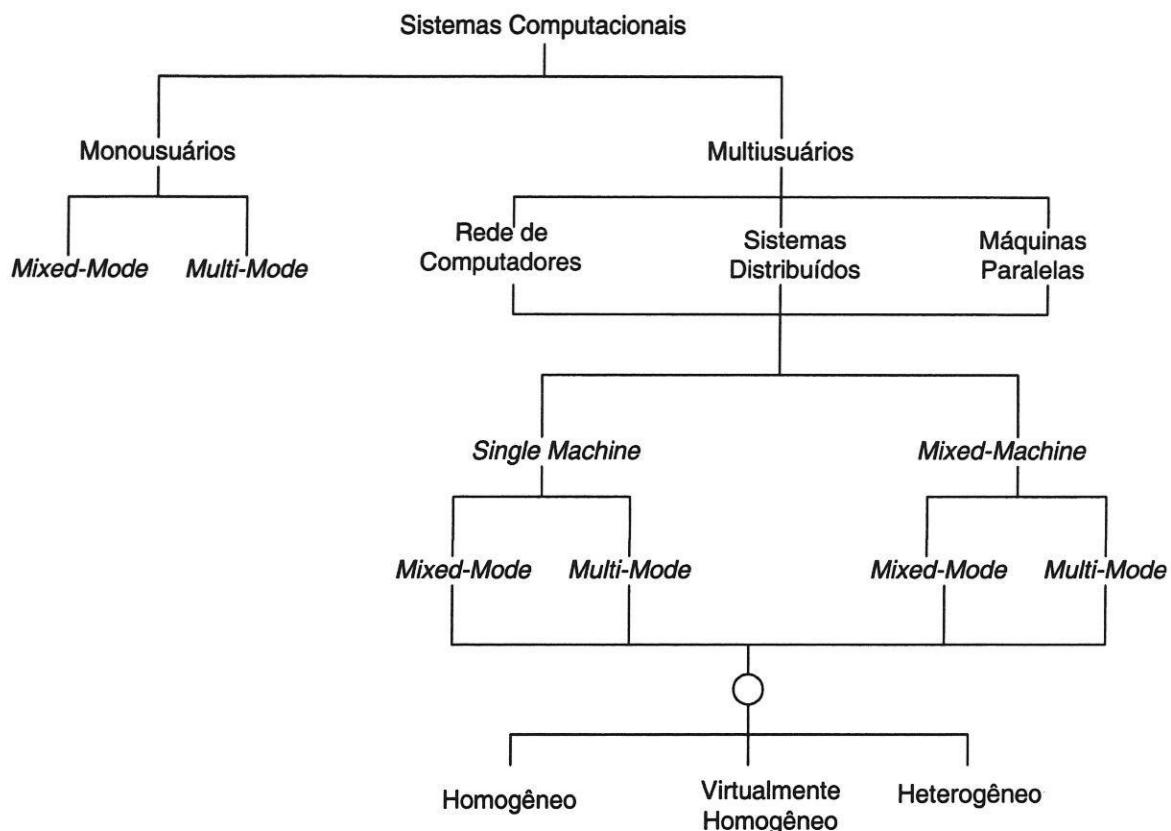


Figura 3 - Taxonomia de heterogeneidade

Tomando como parâmetro a Computação Heterogênea definida por Freund (Freund, 1989) pode-se definir mais precisamente o tipo de análise de código a ser efetuado em termos de particionamento e modularização das tarefas (ou aplicações) a serem mapeadas no sistema.

Quando o sistema for considerado como homogêneo, o particionamento e o mapeamento farão uso de técnicas mais simples como o *round-robin* e o particionamento através da indivisibilidade das tarefas. Por outro lado, quando o sistema for considerado heterogêneo, técnicas de particionamento e modularização de tarefas mais apurados deverão ser utilizadas.

Definidos os termos arquiteturais e os modos de execução encontrados no sistema a ser analisado, as estratégias de mapeamento e escalonamento, bem como as características das aplicações a serem mapeadas podem ser descritas segundo apresentado na taxonomia proposta por (Braun et al., 1998). Dessa maneira, as heurísticas de mapeamento (Kafil & Ahmad, 1997) e a obtenção das sub-tarefas podem ser melhor escolhidas.

Essa nova taxonomia absorve os conceitos presentes nas demais taxonomias, uma vez que pode ser observado:

- modo de execução: tipo de paralelismo presente (síncrono e /ou assíncrono);
- modelo de máquina: arquitetura apresentada pelas máquinas que compõem o sistema (tanto em nível arquitetural quanto de diferenças de *clock*);
- plataforma utilizada: composição do sistema a ser observado (sistemas distribuídos, redes de computadores, máquinas intrinsecamente paralelas);
- estratégia de mapeamento: uma vez que se tem a composição arquitetural do sistema e do modo de execução presente na plataforma utilizada a estratégia de mapeamento a ser utilizada pode ser a mesma apresentada em (Kafil & Ahmad, 1997) e da obtenção das sub-tarefas apresentada por (Braun et al., 1998) e por ;
- grau de heterogeneidade: sabendo-se o quão heterogêneo é o sistema a estratégia de mapeamento torna-se mais flexível permitindo que essa alocação seja feita não somente pelo modo de execução e pela composição arquitetural mas também pela capacidade individual dos recursos que compõem o sistema.

4. Considerações Finais

Este relatório apresentou as principais taxonomias de heterogeneidade existentes na literatura fazendo uma análise crítica de cada uma delas. Apesar do fato dessas taxonomias serem importantes, não contemplam todos os aspectos existentes na composição dos sistemas presentes na computação heterogênea.

Como já mencionado essas taxonomias abordam basicamente a identificação de um sistema no aspecto de ser ou não heterogêneo quanto a arquitetura e o modo de execução presente nas máquinas. Contudo, nenhuma das taxonomias leva em consideração o grau de heterogeneidade apresentado pelo sistema e pelos diversos recursos que compõem esse sistema.

Esse grau de heterogeneidade permite identificar uma heterogeneidade independente de arquitetura e do modo de execução, permite identificar uma

heterogeneidade temporal. Por outro lado, se o sistema é classificado como heterogêneo essa estratégia de mapeamento deve ser mais apurada e sofisticada levando em conta a potencialidade de cada recurso presente no sistema.

Quando o sistema for considerado virtualmente heterogêneo, implica que o fato de tratá-lo como homogêneo ou como heterogêneo (apesar de apresentar certo grau de heterogeneidade) não impõem sobrecarga à estratégia de mapeamento.

Fazendo uso do grau de heterogeneidade e das taxonomias presentes na literatura, uma nova taxonomia foi proposta como forma de extensão e completude das demais existentes (Branco et. al., 2003a) (Branco et al, 2003b).

A taxonomia proposta é concisa e busca abordar todos os aspectos pertinentes a computação heterogênea: modo de execução, modelo de máquina, estratégia de mapeamento e subdivisão de tarefas, plataforma utilizada e grua de heterogeneidade.

Essa nova proposta é mais completa que as anteriormente apresentadas, além de permitir que as idéias sejam mais facilmente organizadas pela sua simplicidade.

Referências Bibliográficas

- Ambrosius, S. L.; Freund, R. F.; Scott, S. L.; Siegel, H. J. (1996). Work-Based Performance Measurement and Analysis of Virtual Heterogeneous Machines. In *the 5th Heterogeneous Computing Workshop (HCW'96)*. p 669-385, April.
- Branco, K. R. L. J. C., Santana, M.J., Santana, R. H. C. (2003a). A Novel Performance Metric for Evaluation of Computer System Heterogeneity. *Proceedings of The International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'2003)*. p. 292 – 301, V. 34, n. 04 SCS. Montreal – Canadá, July.
- Branco, K. R. L. J. C., Santana, M.J., Santana, R. H. C. (2003b). A Novel Metric for Checking Levels of Heterogeneity in Distributed Computer Systems. *Proceedings of The Fourth Congress of Logic Applied to Technology (LAPTEC'2003)*. p. 148 – 155, V. 101, IOS Press. Marília – São Paulo – Brazil, Novembro.
- Braun, T. D; Siegel, H. J.; Beck, N.; Boloni, L.; Maheswaran, M. ; Reuther, A. I.; Robertson, J. P.; Theys, M. D.; Yao, B (1998). A Taxonomy for Describing

- Matching and Scheduling Heuristics for Mixed-Machine Heterogeneous Computing Systems. *IEEE Workshop on Advances in Parallel and Distributed Systems* p. 330-335, October.
- Casavant; T.L.; Kuhl, J.G. (1988). A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems. *IEEE Transactions on Software Engineering*, p.141-154, February.
- Colouris, G., Dollimore, J., Kindberg, T.(1994). *Distributed Systems Concepts and Design*, 2^a ed., Addison-Wesley Publishing Company, s.1.
- Duncan, R. (1990) A Survey of Parallel Computer Architectures. *IEEE Computer*, p. 5-16.
- Ekemecic, I; Tartalja, I.; Milutinovic, V. (1996). A Survey of Heterogeneous Computing: Concepts and Systems. *Proceedings of IEEE*, 84: 1127-1144.
- El-Rewini, H.; Ali, H. H.; Lewis, T. (1995). Task Scheduling in Multiprocessing Systems. *IEEE Computer*, p. 27-37, December.
- Eshaghian, M.M.(1996). *Heterogeneous Computing*. Artech House, Norwood, MA.
- Flynn, M. J. (1996). Very High-Speed Computing Systems. *Proceedings of IEEE*. Vol. 54, nº 12, p. 1901-1909, December.
- Freund, R.F. (1989). Optimal selection theory for superconcurrency. *Proceedings of Supercomputing '89*, p. 699-703, November.
- Freund, R.; Conwell, D. (1990). Superconcurrency: A Form of Distributed Heterogeneous Supercomputing. *Supercomputing Review*, p. 47-50, October.
- Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; Sunderam, V. (1994). PVM: Parallel Virtual Machine. *A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, Londres.
- Ibaraki, T. (1987). Combinatorial Optimization Problems And Their Complexity, In: *Enumerative Approaches To Combinatorial Optimization - Part I. Annals of Operations Research*, v. 10.
- Kafil, M.; Ahmad, I. (1997). Optimal Task Assignment in Heterogeneous Computing Systems. *6th Heterogeneous Computing Workshop (HCW'97)*, p. 135-146, April.
- Khokhar, A. A.; Prasanna, V.K.; Shaaban, M.E.; Wang, C.L. (1993). Heterogeneous Computing: Challenges and Opportunities. *IEEE Computer*, 26(6): 18-27, June.

Santana, M.J.; Santana, R.H.C.; Francês, C. R. L.; Orlandi, R.C. (1997). Tools and Methodologies For Performance Evaluation of Distributed Systems – A Comparison Study. *In The Proceedings of the: Summer Computer Simulation Conference*, Arlington, Virginia. Proceedings. Arlington, p. 124-28.

Watson, D. W.; Siegel, H. J.; Antonio, J. K.; Nichols, M. A.; Atallah, M. J. (1993). A framework for compile-time selection of parallel modes in a SIMD/SPMD heterogeneous environment. *2nd Workshop on Heterogeneous Processing (WHP'93)* p. 57-64, April.