

Arquitetura de Computadores III

Arquiteturas paralelas, classificações, clusters e supercomputadores

Arquiteturas monoprocessadas

- Os processos compartilham o mesmo processador.
- S.O. pode ser implementado com o conceito de monoprogramação ou multiprogramação.

Arquiteturas monoprocessadas

- Monoprogramação: recursos do computador alocados para uma única tarefa até o seu término.
- Multiprogramação: processador alterna a execução de vários processos.

Arquiteturas multiprocessadas

- Multiprocessada: vários elementos de processamento.
- Tipos de arquiteturas multiprocessadas:
 - Memória compartilhada
 - Memória distribuída

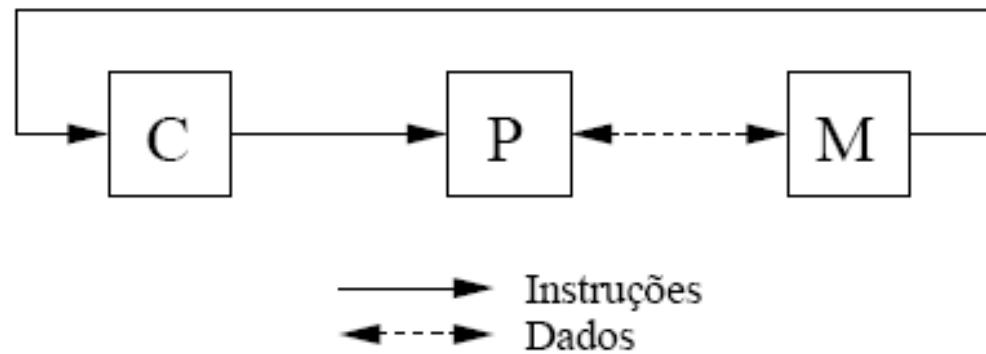
Classificações de Flynn

Classificação de Flynn (Flynn, 1972) segundo o fluxo de instruções e fluxo de dados.

	SD (Single Data)	MD (Multiple Data)
SI (Single Instruction)	SISD Máquinas von Neumann	SIMD Máquinas Array
MI (Multiple Instruction)	MISD Sem representante até agora	MIMD Multiprocessadores e multicamputadores

De Rose, Navaux, Arquiteturas Paralelas (2003)

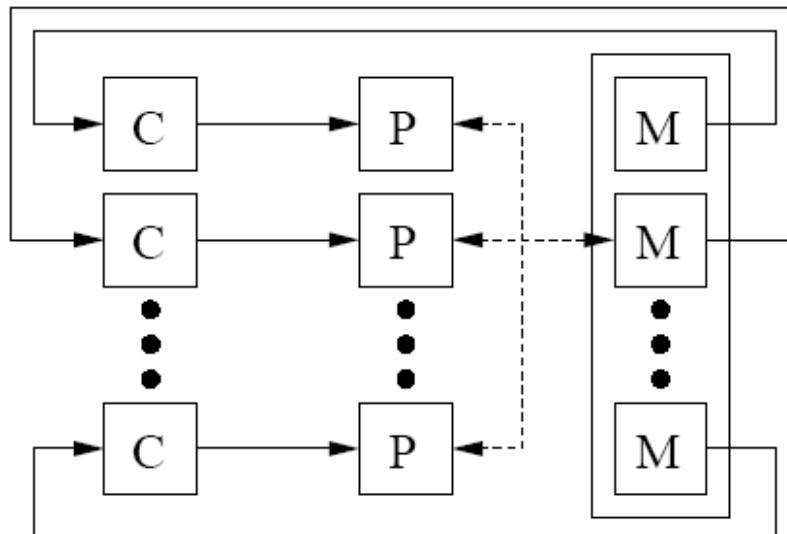
Classificações de Flynn



SISD

De Rose, Navaux, Arquiteturas Paralelas (2003)

Classificações de Flynn

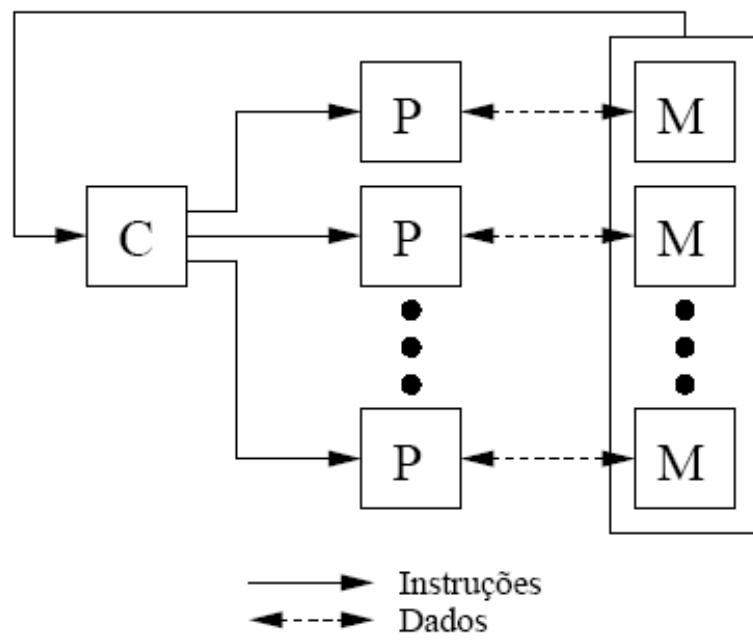


→ Instruções
← → Dados

De Rose, Navaux, Arquiteturas Paralelas (2003)

MISD

Classificações de Flynn

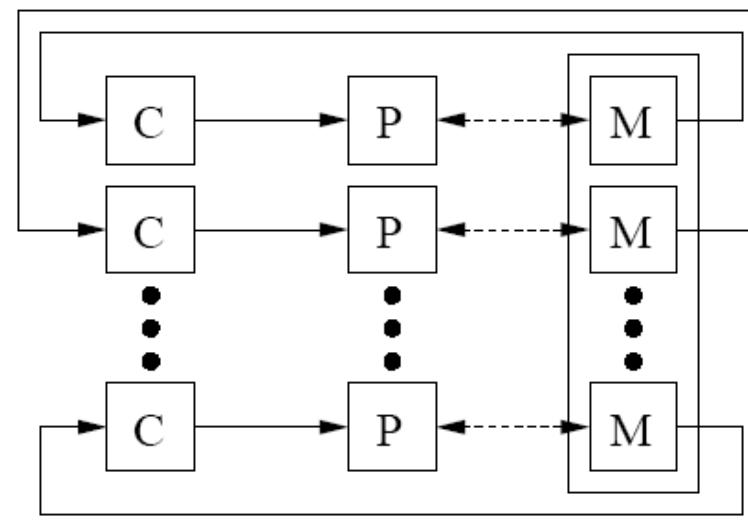


SIMD

- Processadores vetoriais:
 - Vetor é um conjunto de dados escalares do mesmo tipo, armazenados em memória.
 - Processamento vetorial ocorre quando executamos operações aritméticas ou lógicas sobre vetores.
 - Um processador escalar opera sobre um ou um par de dados.

De Rose, Navaux, Arquiteturas Paralelas (2003)

Classificações de Flynn

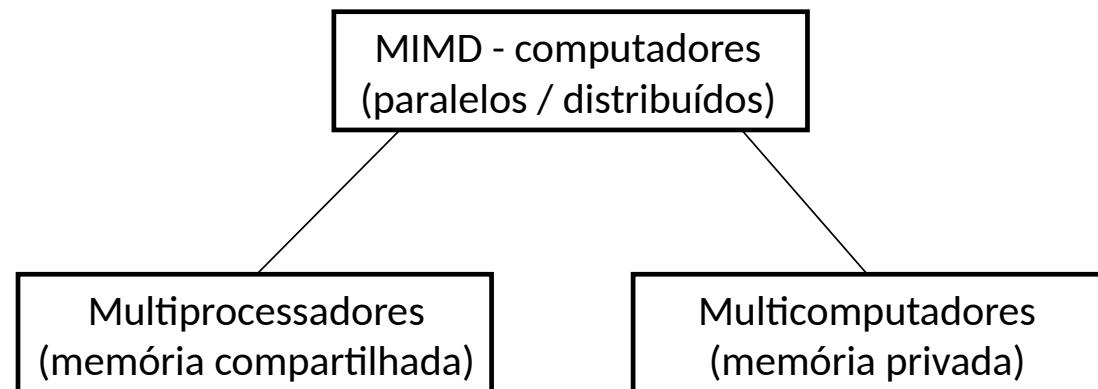


→ Instruções
← → Dados

De Rose, Navaux, Arquiteturas Paralelas (2003)

MIMD

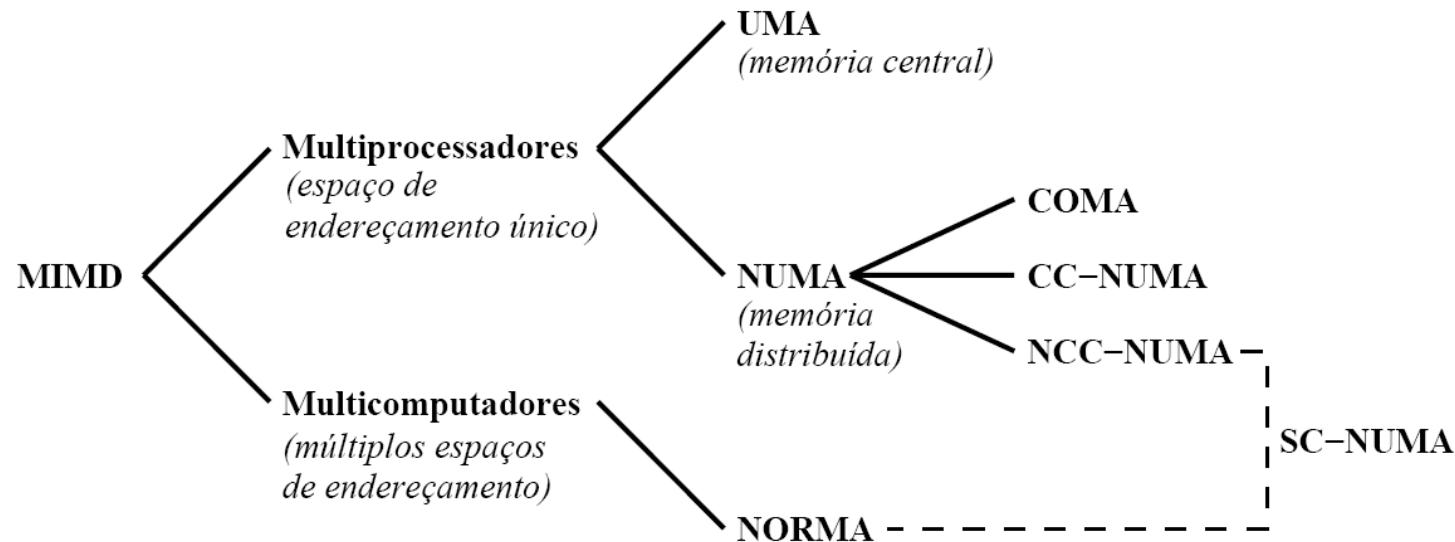
Subdivisão da classe MIMD



Classificações segundo compartilhamento de memória

- UMA: Uniform Memory Access
- NUMA: Non-Uniform Memory Access
- CC-NUMA: Cache-Coherent NUMA
- NCC-NUMA: Non-Cache-Coherent NUMA
- SC-NUMA: Software-Coherent NUMA
- COMA: Cache-Only Memory Architecture
- NORMA: Non-Remote Memory Access

Classificações segundo compartilhamento de memória



De Rose, Navaux, Arquiteturas Paralelas (2003)

A linha tracejada indica que através de um software que implemente coerência de *cache*, as máquinas NCC-NUMA e NORMA podem se transformar em máquinas SC-NUMA.

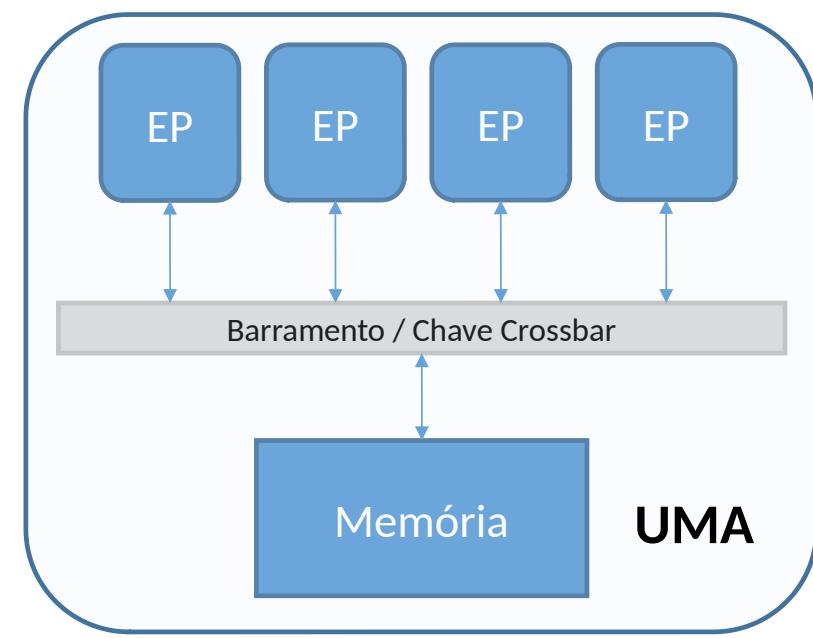
Memória Compartilhada

- Elementos de processamento compartilham a mesma memória.
- Programação realizada através de variável compartilhada. Maior facilidade na construção de programas paralelos.
- Neste tipo de arquitetura existe uma limitação de número de nós.
- Escalabilidade não é total. **Por que?**
- Programação em memória compartilhada é realizada com threads.
- Exemplos:
 - Pthreads
 - OpenMP

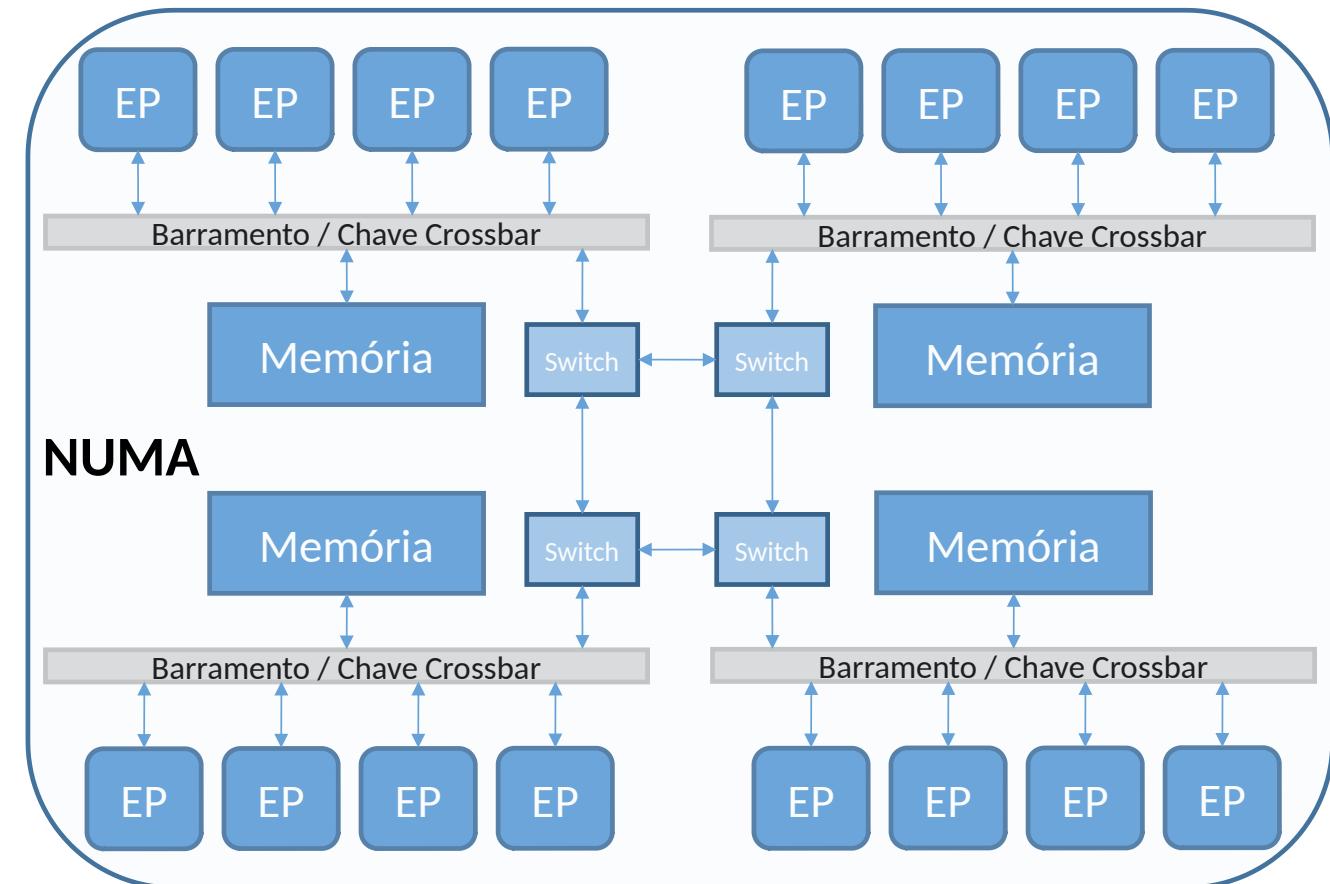
Memória Compartilhada

- O desempenho neste tipo de sistema é maior se consideramos no seu projeto o uso de memória cache.
- Surge o problema de coerência de cache.
- Solução em software ou em hardware.

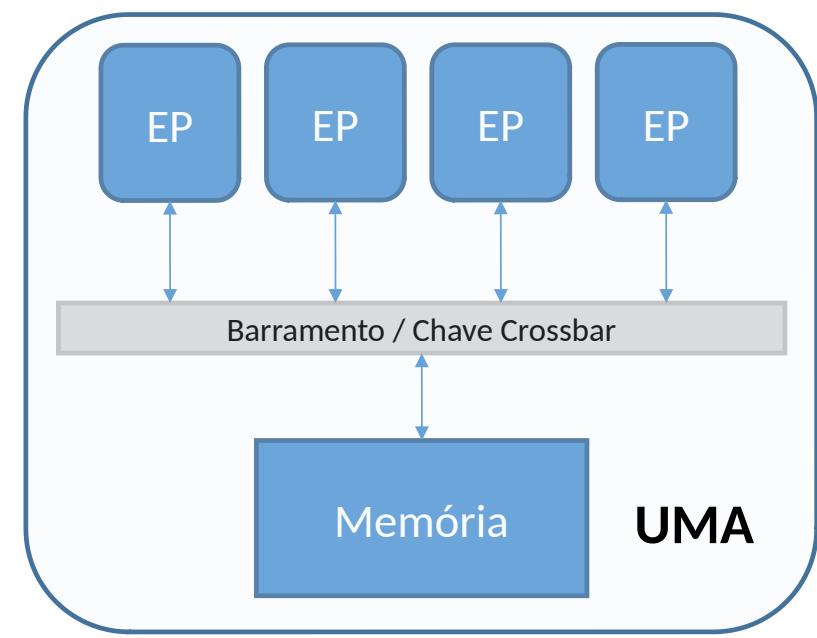
UMA vs. NUMA



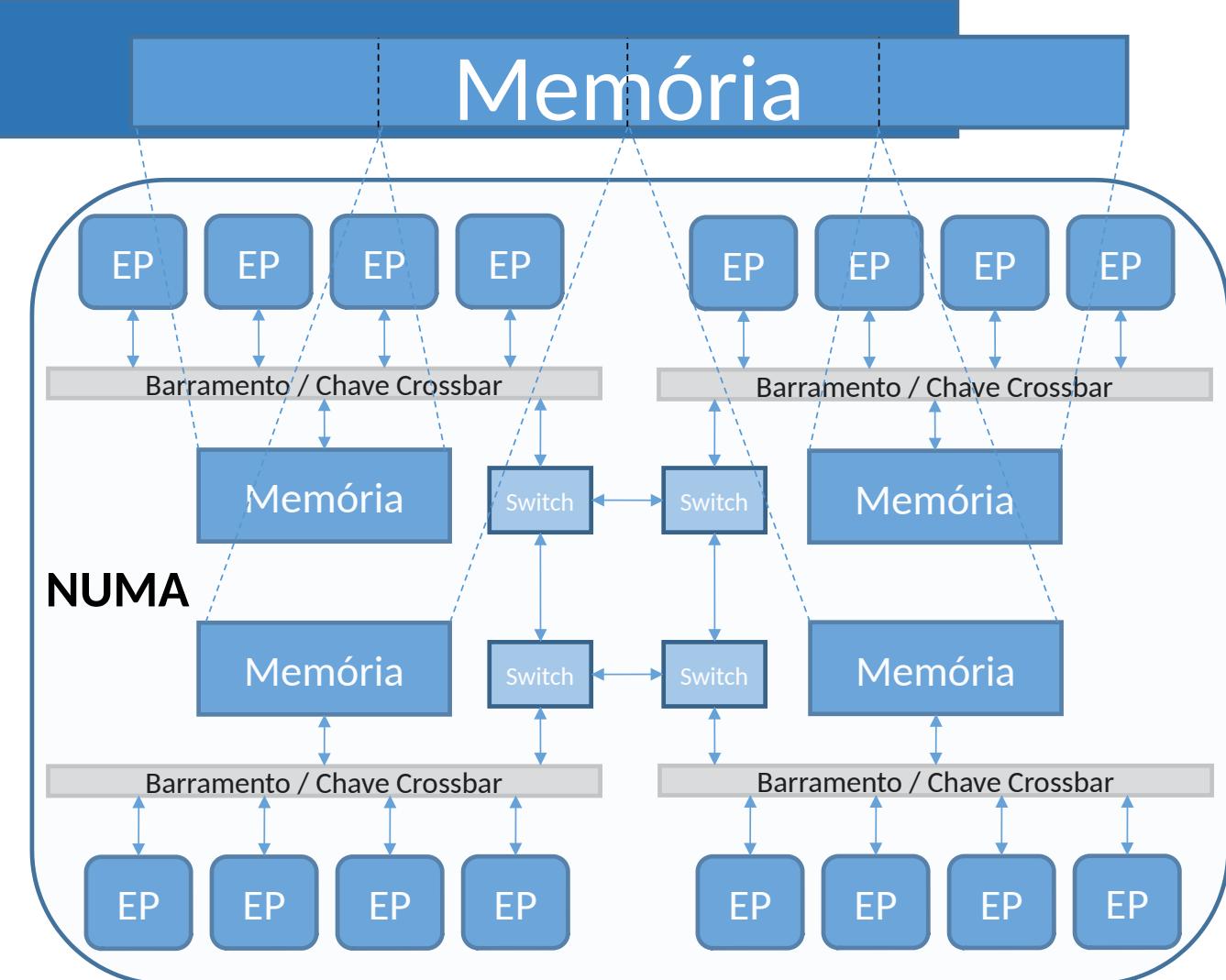
EP: Elemento de Processamento



UMA vs. NUMA

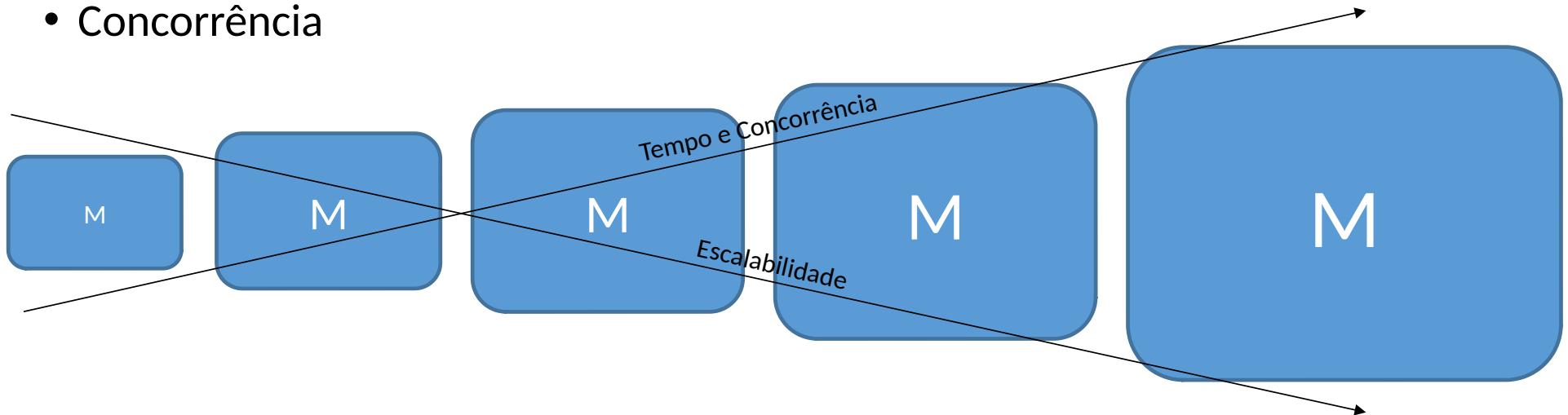


EP: Elemento de Processamento



Qual a vantagem de usar uma máquina NUMA?

- Restrições existentes em relação à memória (M) centralizada que podem afetar o desempenho e o consumo de energia:
 - Escalabilidade
 - Tempo de acesso
 - Concorrência

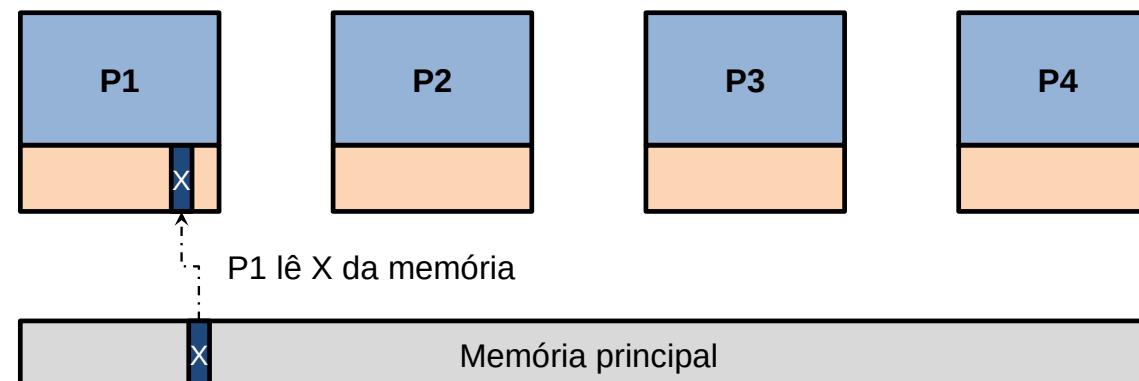


Uso compartilhado da memória

- A maioria dos processadores atuais possuem múltiplos núcleos
- Diversos núcleos compartilham o acesso a dados em uma mesma unidade de memória
- Outro problema: garantir a **coerência** do dado, o que é resolvido com algoritmos específicos para este fim

Problema da coerência

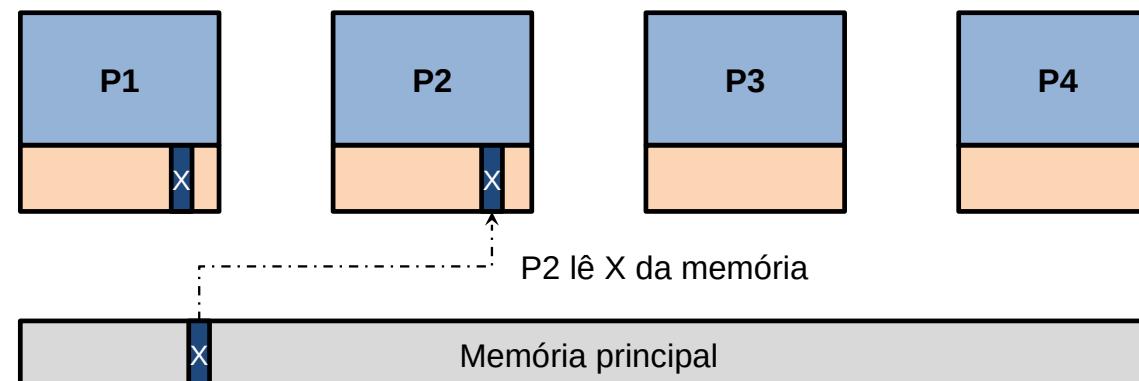
- Multicores: Múltiplos processadores e caches
- Múltiplas cópias dos mesmos dados em caches diferentes



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

Problema da coerência

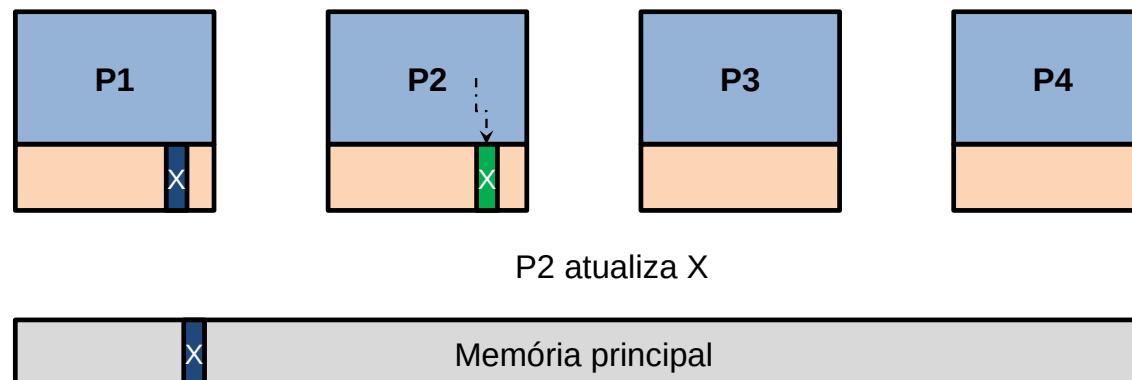
- Multicores: Múltiplos processadores e caches
- Múltiplas cópias dos mesmos dados em caches diferentes



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

Problema da coerência

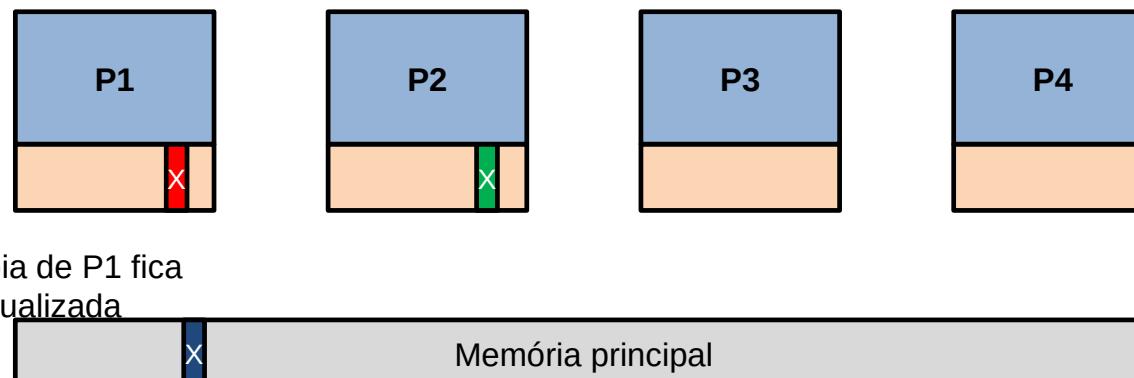
- Multicores: Múltiplos processadores e caches
- Múltiplas cópias dos mesmos dados em caches diferentes



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

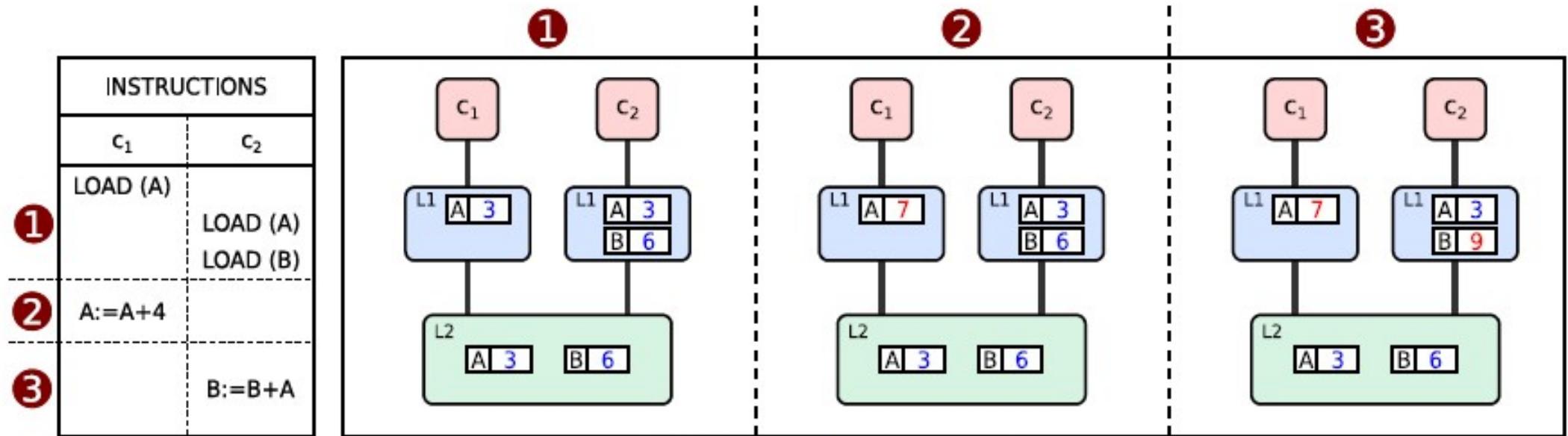
Problema da coerência

- Multicores: Múltiplos processadores e caches
- Múltiplas cópias dos mesmos dados em caches diferentes



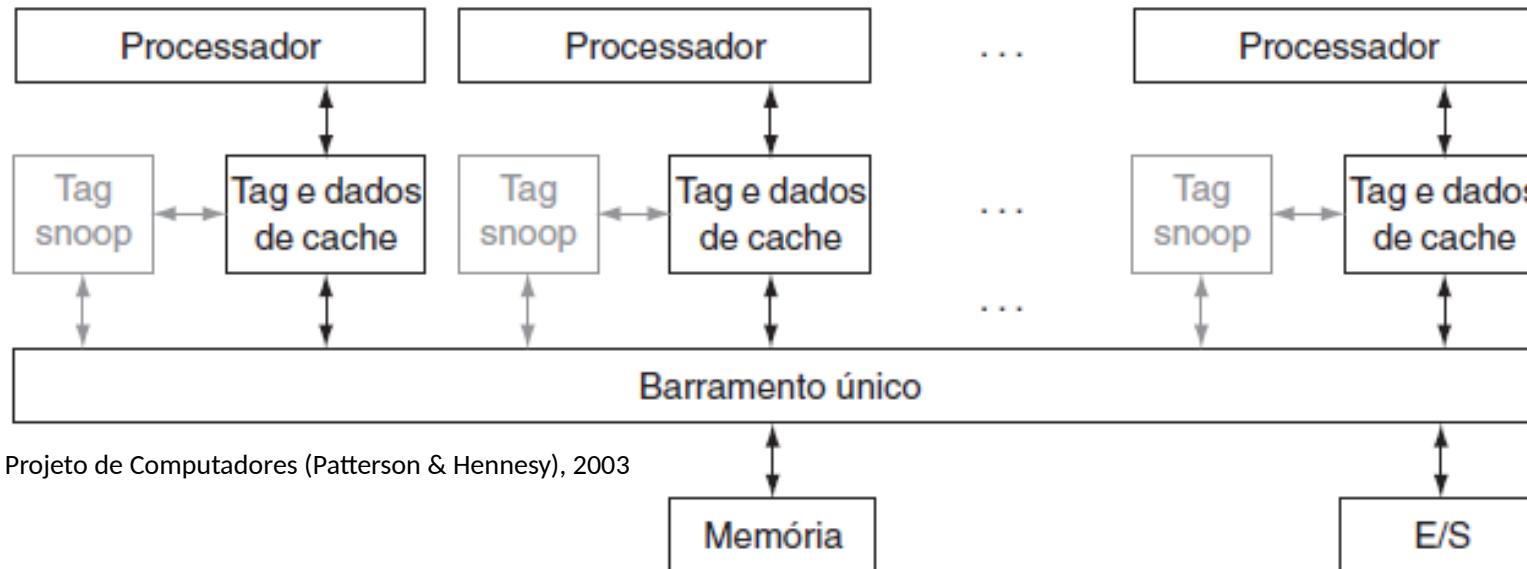
Souza, M. A., *Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning*,
Tese, PUC Minas, 2021

Problema da coerência



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

Coerência de cache (Snooping)



Organização e Projeto de Computadores (Patterson & Hennesy), 2003

Coerência de cache com snooping. Um método para manter coerência de cache em que todos os controladores de cache monitoram o barramento para determinar se possuem ou não uma cópia do bloco desejado.

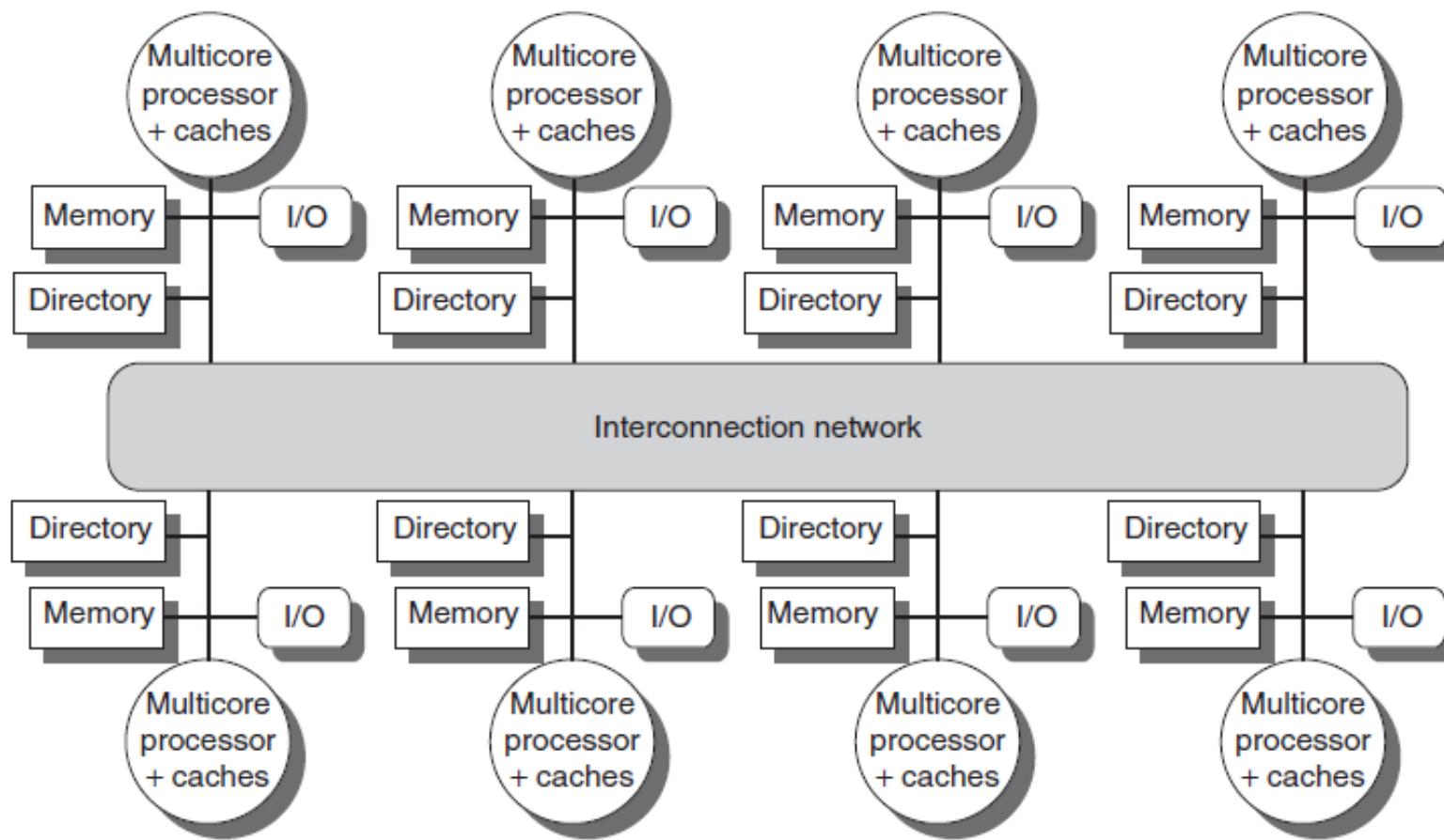
Write-invalidate. Um tipo de protocolo de snooping em que o processador de escrita faz com que todas as cópias em outras caches sejam invalidadas antes de mudar sua cópia local, o que permite atualizar os dados locais até que outro processador os solicite.

Write-update. Um tipo de protocolo de snooping em que o processador de escrita faz com que todas as cópias em outras caches sejam atualizadas.

Coerência de cache (Diretório)

- Cada módulo de memória possui um diretório local que armazena as informações sobre onde as cópias dos blocos estão residentes.
- Os protocolos baseados em diretório enviam comandos de consistência seletivamente para aquelas caches que possuem uma cópia válida do bloco de dados compartilhado.

Coerência de cache (Diretório)

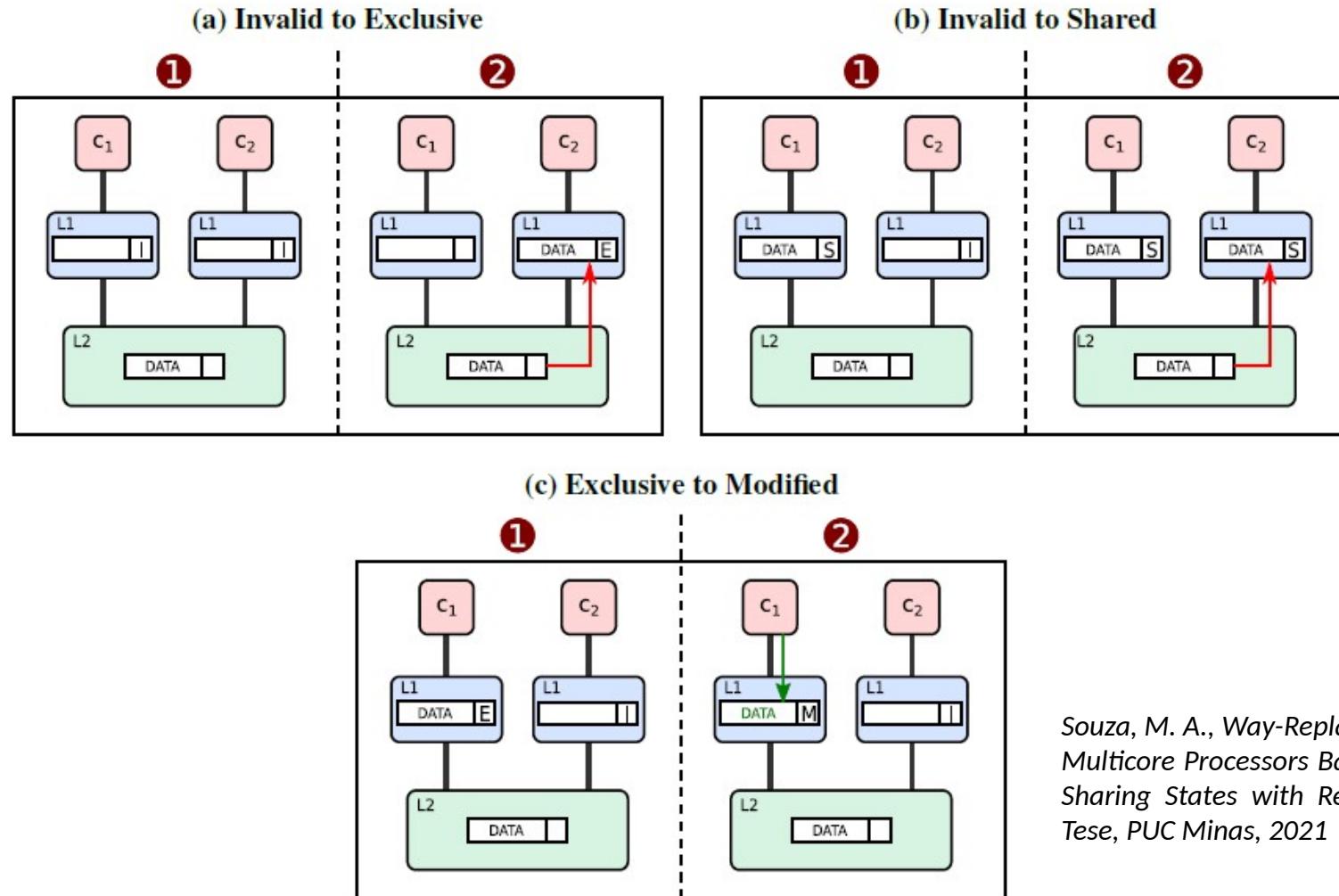


HENNESSY, John L.,
PATTERSON, David A.,
Computer Architecture
A Quantitative Approach

Coerência de cache (MSI, MESI, MOESI)

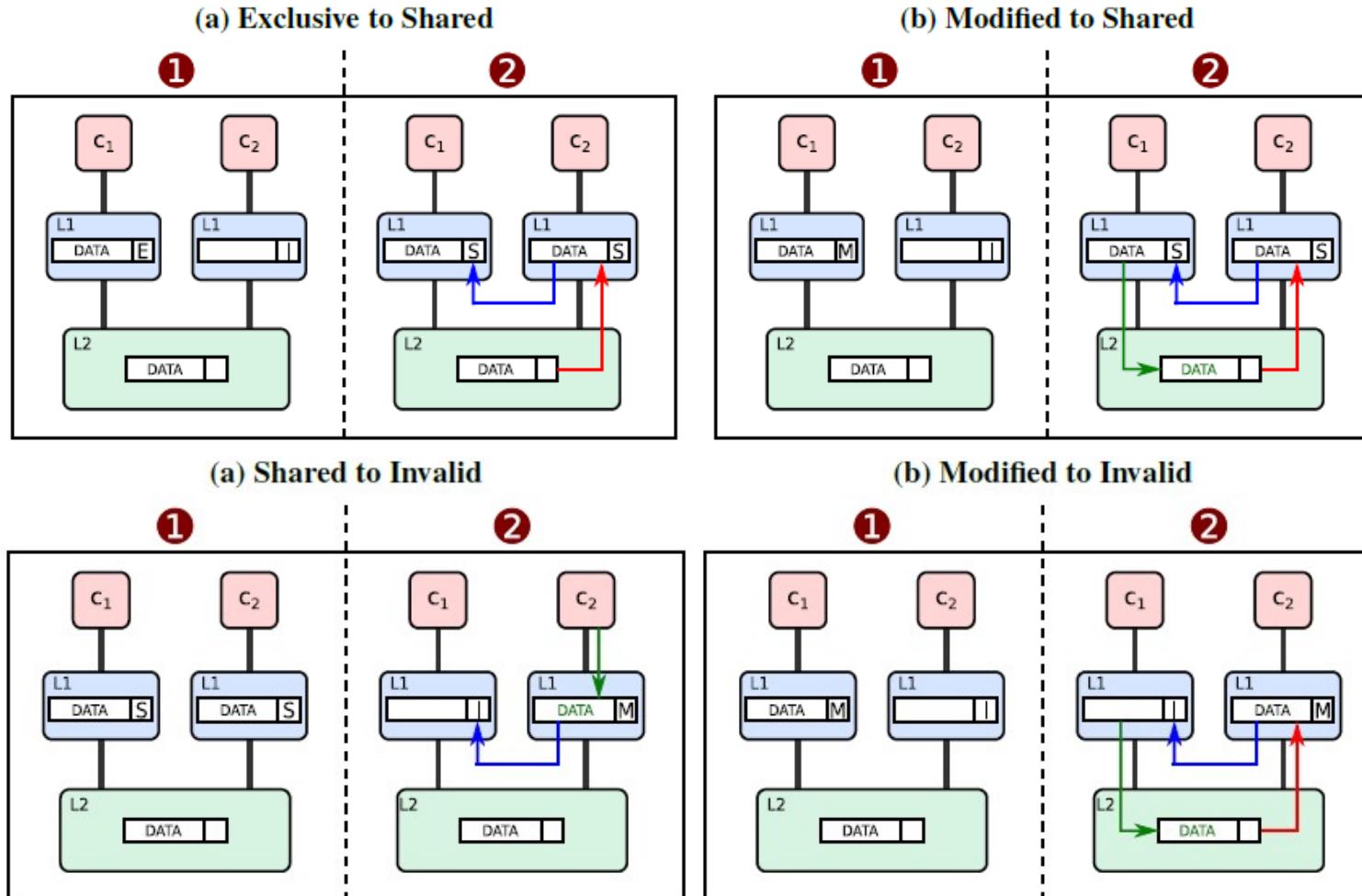
- Estados que cada bloco na memória cache possui:
 - Inválido (I): bloco inválido na memória cache.
 - Shared (S) ou Compartilhado: bloco só foi lido e pode haver cópias em outras memórias cache.
 - Modificado (M): apenas essa cache possui cópia do bloco e a memória principal não está atualizada.
 - Exclusivo (E): Apenas essa cache possui cópia do bloco e a memória principal está atualizada.
 - Owner (O) : Essa cache supre o dado em caso de leitura com falha no barramento uma vez que a memória não está atualizada. Outras caches podem ter cópia do dado.

Protocolo MESI



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

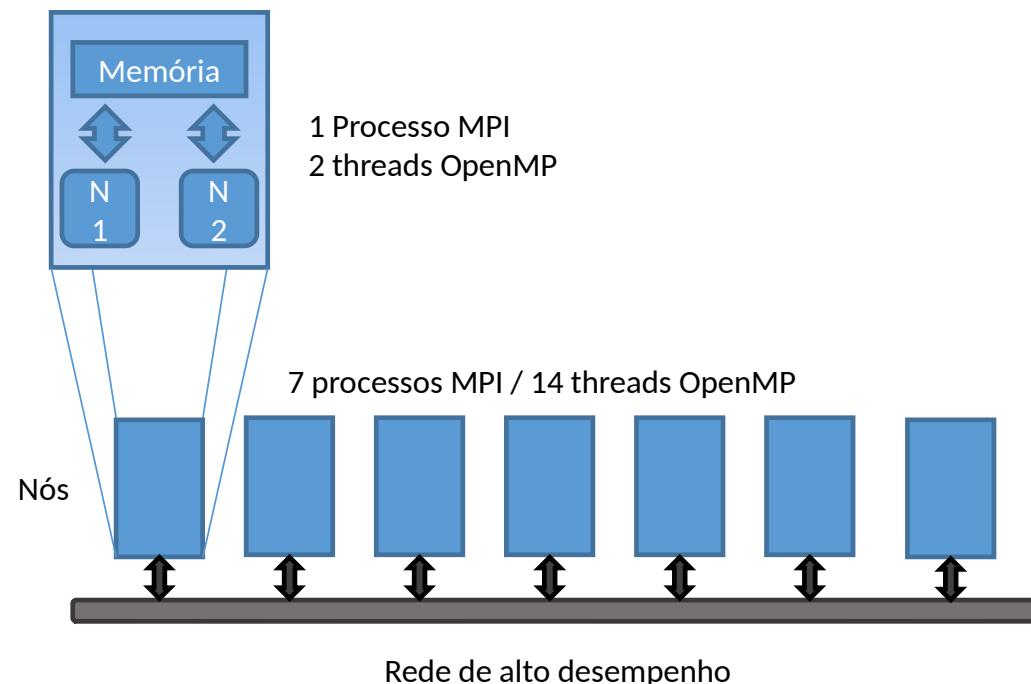
Protocolo MESI



Souza, M. A., Way-Replacement Algorithms for Multicore Processors Based on Coherence and Sharing States with Reinforcement Learning, Tese, PUC Minas, 2021

Memória Distribuída

- Grupo de computadores autônomos (nós) que trabalham juntos como um recurso único.
- Os nós são interconectados através de redes de alto desempenho.



Memória Distribuída

- Escalabilidade absoluta e incremental.
- Alta disponibilidade.
- Excelente custo benefício.
- Comunicação realizada através de passagem de mensagens.
 - MPI (Message Passing Interface) ou
 - PVM (Parallel Virtual Machine).

Memória Distribuída

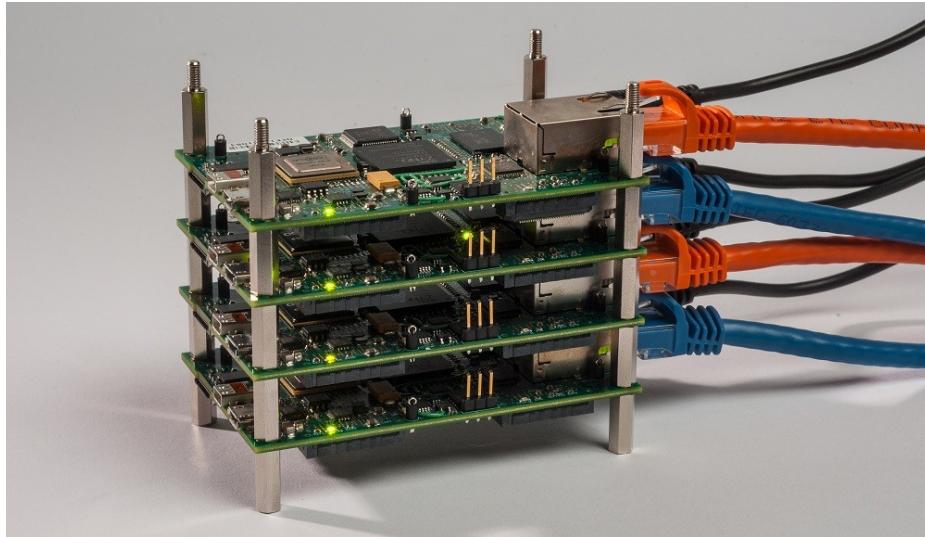
- Cluster ou aglomerado de computadores.
 - São usados em gerenciadores de bancos de dados, com servidores WEB.
 - São usados principalmente com processamento paralelo.
- Grids ou grades computacionais.

Cluster de Computadores

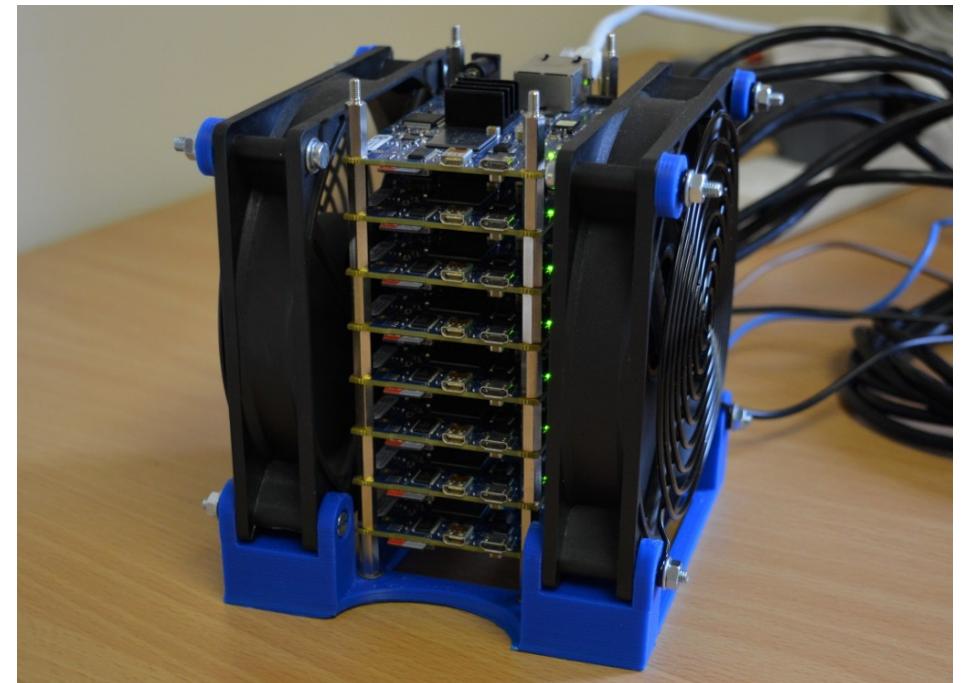


<https://hpcf.umbc.edu/system-description-maya/>

Cluster Parallelia Board

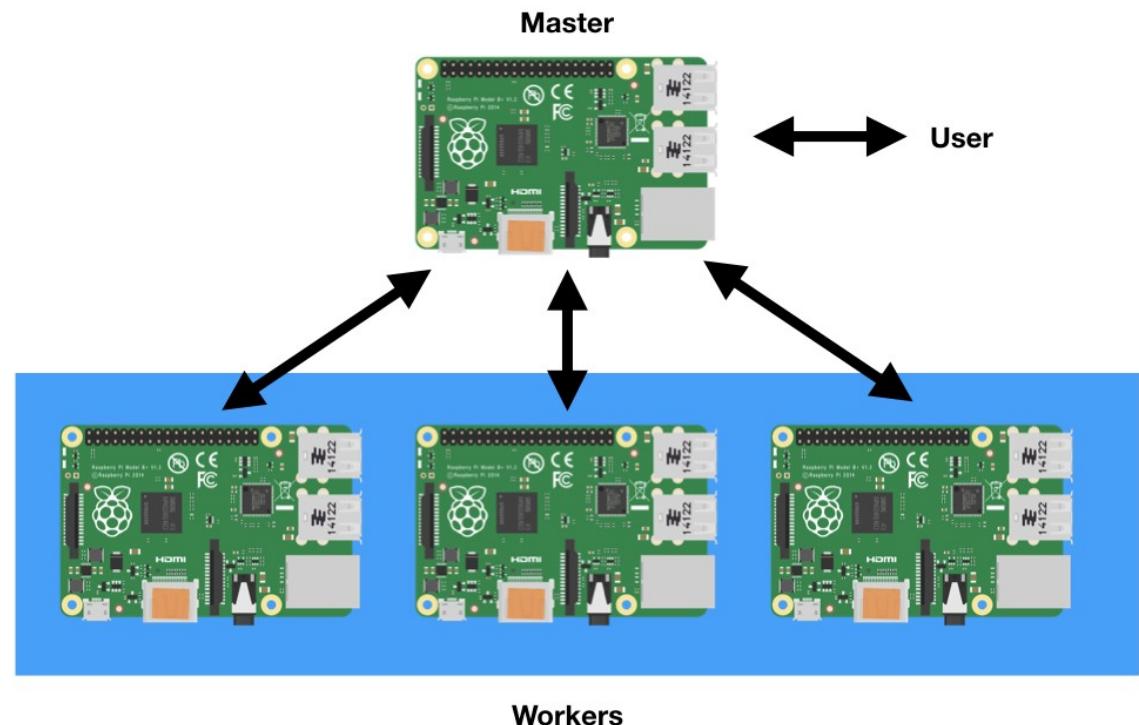
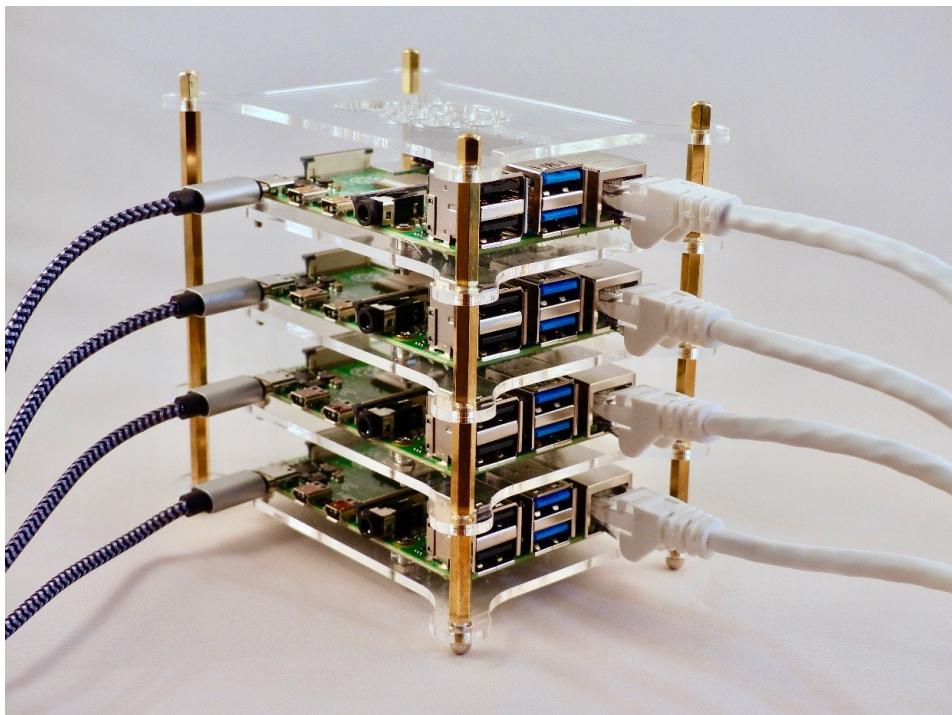


<https://www.parallelia.org/2013/08/21/parallelia-hardware-update/>



<https://www.parallelia.org/2014/04/30/cases-and-cooling/>

Cluster Raspberry Pi 4



<https://magpi.raspberrypi.org/articles/build-a-raspberry-pi-cluster-computer>

Cluster Raspberry Pi 2 Model B



Photo courtesy of CArT

Jaguar - Cray XT5-HE Opteron Six Core 2.6 GHz



<https://phys.org/news/2009-11-oak-ridge-jaguar-supercomputer-world.html>

Clusters de Computadores (top500.org)

Rank	Site	Computer/Year Vendor	Cores	R _{max}	R _{peak}	Power	
1	Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.60	MPP
2	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband / 2009 IBM	122400	1042.00	1375.78	2345.50	Cluster
3	National Institute for Computational Sciences/University of Tennessee United States	Kraken XT5 - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009 Cray Inc.	98928	831.70	1028.85		MPP
4	Forschungszentrum Juelich (FZJ) Germany	JUGENE - Blue Gene/P Solution / 2009 IBM	294912	825.50	1002.70	2268.00	MPP
5	National SuperComputer Center in Tianjin/NUDT China	Tianhe-1 - NUDT TH-1 Cluster, Xeon E5540/E5450, ATI Radeon HD 4870 2, Infiniband / 2009 NUDT	71680	563.10	1206.19		Cluster
6	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix ICE 8200EX, Xeon QC 3.0 GHz/Nehalem EP 2.93 Ghz / 2009 SGI	56320	544.30	673.26	2348.00	MPP
7	DOE/NNSA/LLNL United States	BlueGene/L - eServer Blue Gene Solution / 2007 IBM	212992	478.20	596.38	2329.60	MPP
8	Argonne National Laboratory United States	Blue Gene/P Solution / 2007 IBM	163840	458.61	557.06	1260.00	MPP
9	Texas Advanced Computing Center/Univ. of Texas United States	Ranger - SunBlade x6420, Opteron QC 2.3 Ghz, Infiniband / 2008 Sun Microsystems	62976	433.20	579.38	2000.00	Cluster
10	Sandia National Laboratories / National Renewable Energy Laboratory United States	Red Sky - Sun Blade x6275, Xeon X55xx 2.93 Ghz, Infiniband / 2009 Sun Microsystems	41616	423.90	487.74		Cluster

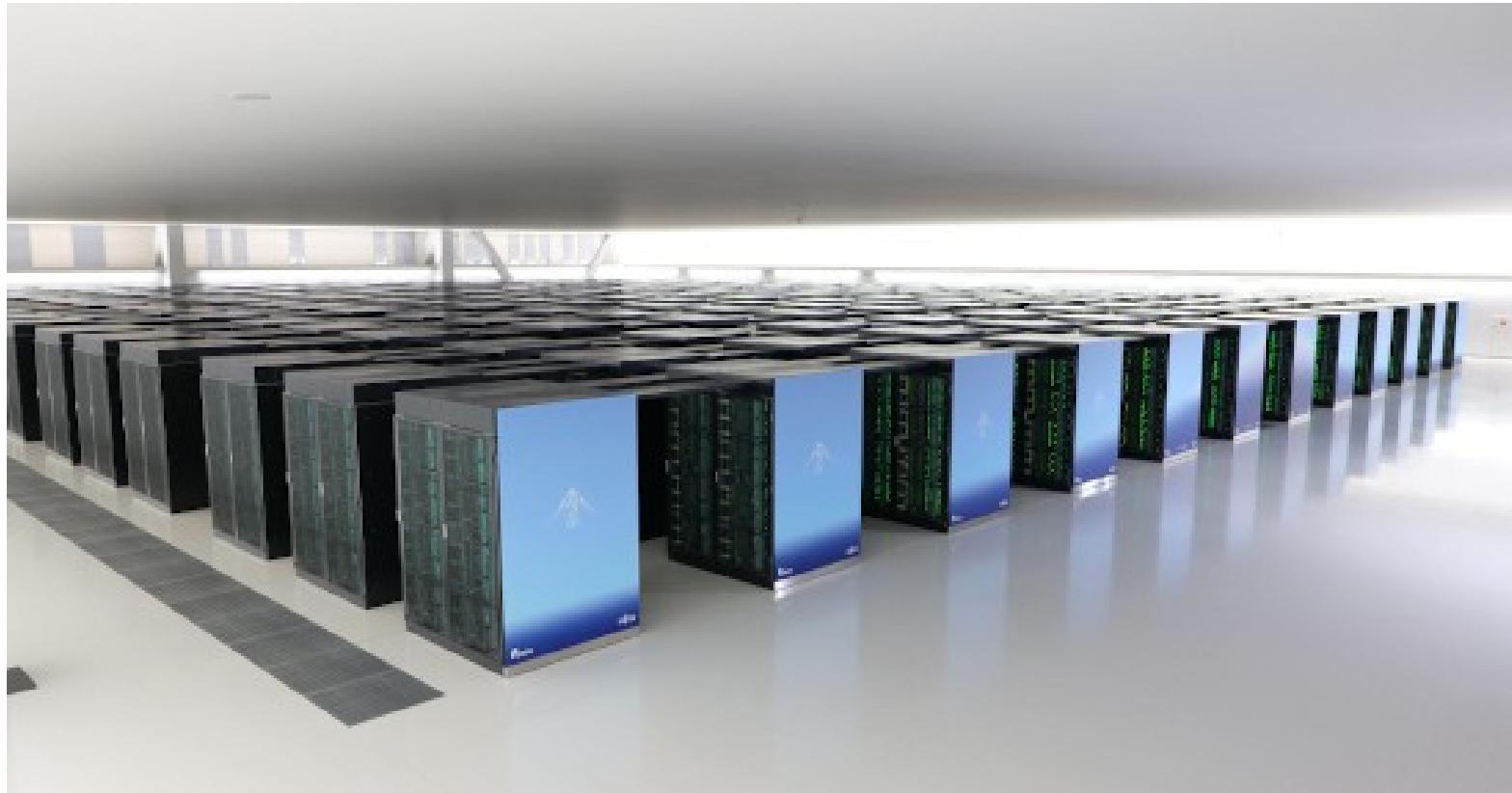
Novembro de 2009

Clusters de Computadores (top500.org)

Rank	Site	Computer
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 Vlllfx 2.0GHz, Tofu interconnect Fujitsu
2	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C NUDT
3	DOE/SC/Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz Cray Inc.
4	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU Dawning
5	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows NEC/HP
6	DOE/NNSA/LANL/SNL United States	Cielo - Cray XE6 8-core 2.4 GHz Cray Inc.
7	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix ICE 8200EX/8400EX, Xeon HT QC 3.0/Xeon 5570/5670 2.93 Ghz, Infiniband SGI
8	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 12-core 2.1 GHz Cray Inc.
9	Commissariat a l'Energie Atomique (CEA) France	Tera-100 - Bull bullex super-node S6010/S6030 Bull SA
10	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband IBM

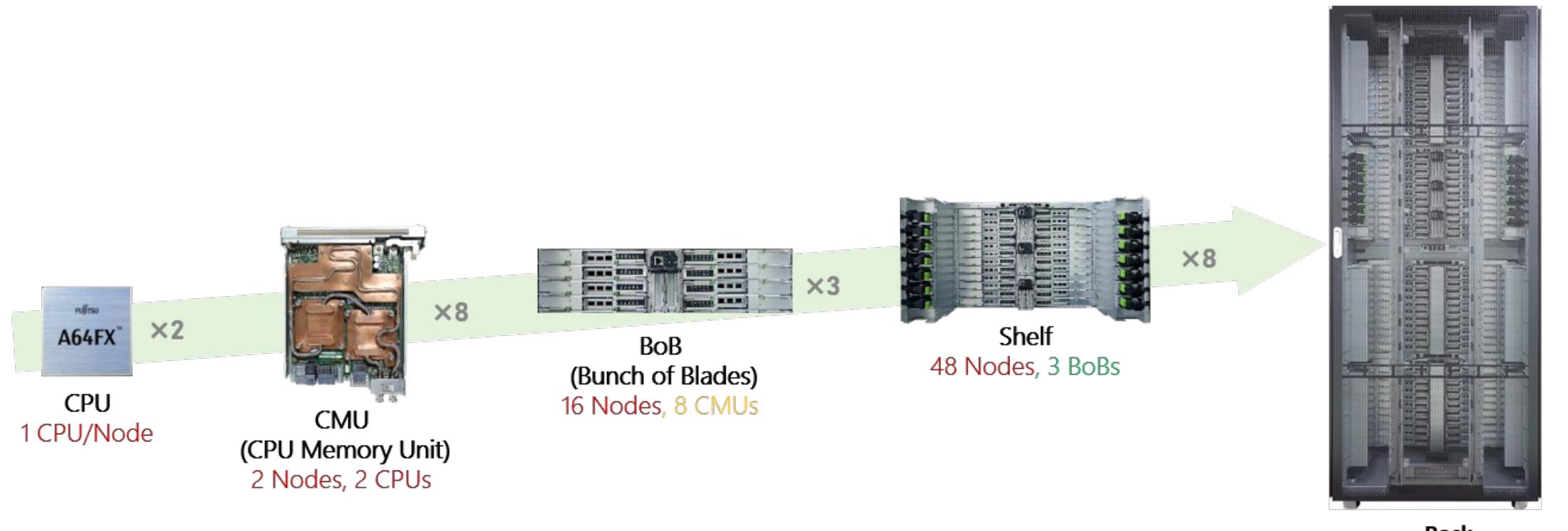
Junho de 2011

Cluster de Computadores



https://www.riken.jp/en/news_pubs/news/2020/20200623_1/

Supercomputador Fugaku (Fujitsu, Japão)



<https://postk-web.r-ccs.riken.jp/spec.html>

Courtesy of FUJITSU LIMITED

297 racks = 7.299.072 núcleos

Clusters de Computadores (top500.org)

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415,530.0	513,854.7	28,335
2	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482

TOP500 LIST - JUNE 2020

Clusters de Computadores (top500.org) - Brasil

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
56	Atlas - Bull 4029GP-TVRT, Xeon Gold 6240 18C 2.6GHz, NVIDIA Tesla V100, Infiniband EDR, Atos Petróleo Brasileiro S.A Brazil	91,936	4,376.0	8,848.5	547
82	Fênix - Bull 4029GP-TVRT, Xeon Gold 5122 4C 3.6GHz, NVIDIA Tesla V100, Infiniband EDR, Atos Petróleo Brasileiro S.A Brazil	60,480	3,161.0	5,371.8	390
240	Santos Dumont (SDumont) - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos Laboratório Nacional de Computação Científica Brazil	33,856	1,849.0	2,727.0	
395	Ogbon Cimatec/Petrobras - Bull Sequana X1000, Xeon Gold 6240 18C 2.6GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos SENAI CIMATEC Brazil	27,768	1,605.0	2,323.3	

TOP500 LIST - JUNE 2020

Clusters de Computadores

Santos Dumont Supercomputer



<https://sdumont.lncc.br/>

Clusters de Computadores (top500.org)

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	Perlmutter - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC	761,856	70,870.0	93,750.0	2,589

TOP500 LIST - NOVEMBER 2021

Clusters de Computadores (top500.org) - Brasil

55	Dragão - Supermicro SYS-4029GP-TVRT, Xeon Gold 6230R 26C 2.1GHz, NVIDIA Tesla V100, Infiniband EDR, Atos Petróleo Brasileiro S.A Brazil	188,224	8,983.0	14,006.5	943
107	Atlas - Bull 4029GP-TVRT, Xeon Gold 6240 18C 2.6GHz, NVIDIA Tesla V100, Infiniband EDR, Atos Petróleo Brasileiro S.A Brazil	91,936	4,376.0	8,848.5	547
125	IARA - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100 SXM4 40 GB, Infiniband, Nvidia SiDi Brazil	24,800	3,657.0	4,130.4	
146	Fênix - Bull 4029GP-TVRT, Xeon Gold 5122 4C 3.6GHz, NVIDIA Tesla V100, Infiniband EDR, Atos Petróleo Brasileiro S.A Brazil	60,480	3,161.0	5,371.8	390
388	Santos Dumont (SDumont) - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos Laboratório Nacional de Computação Científica Brazil	33,856	1,849.0	2,727.0	

TOP500 LIST - NOVEMBER 2021

Lei de Amdahl

- O ganho ideal é limitado pela fração do Código não melhorada.
- O ganho real é a razão dos tempos de execução:

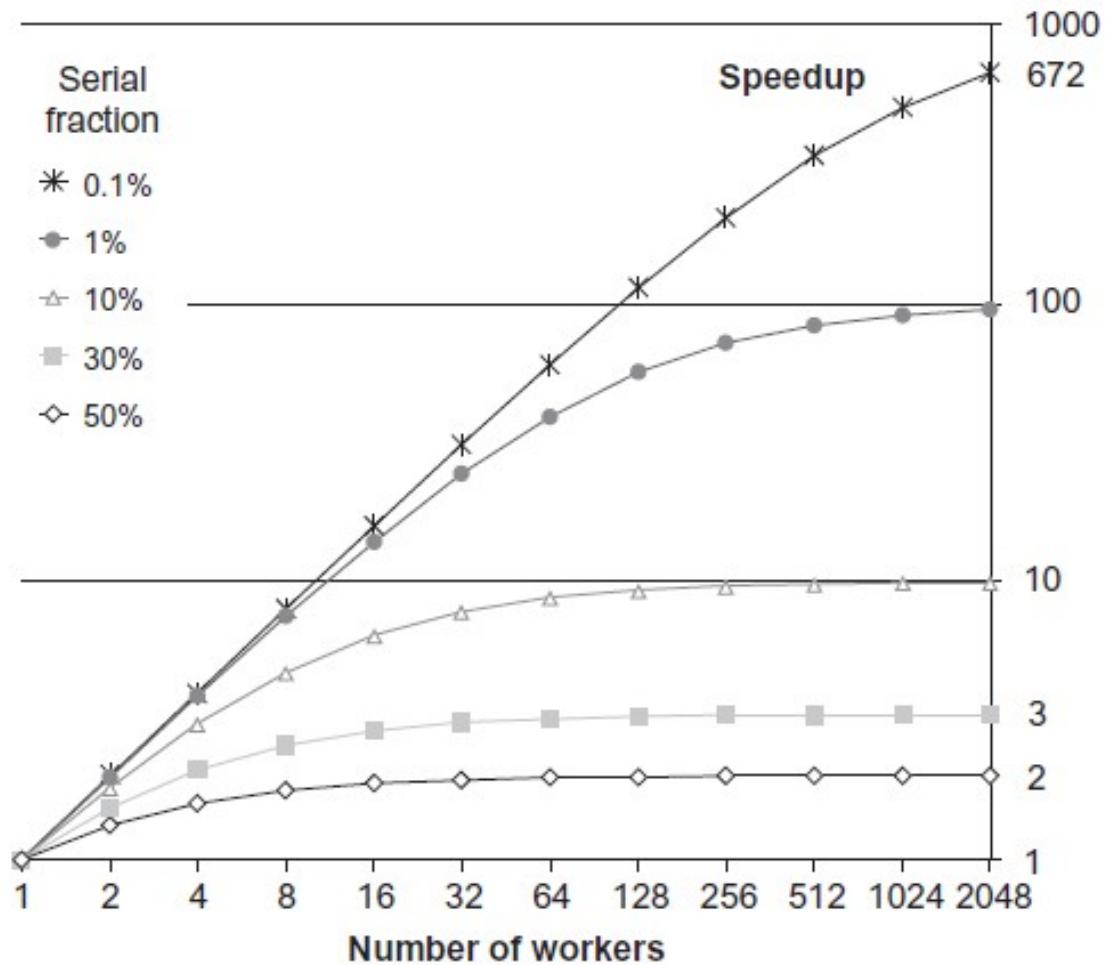
$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

- Suponha que nós queremos melhorar o processador usado para serviço Web. O nome processador é 10 vezes mais rápido em computação do que o processador original. Assumindo que o processador está 40% do tempo ocupado com computação e outros 60% esperando por I/O. Qual é o ganho obtido pelo melhoramento?

$$\text{Fraction}_{\text{enhanced}} = 0.4; \text{Speedup}_{\text{enhanced}} = 10; \text{Speedup}_{\text{overall}} = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} \approx 1.56$$

HENNESSY, John L.,
PATTERSON, David A.,
Computer Architecture
A Quantitative Approach

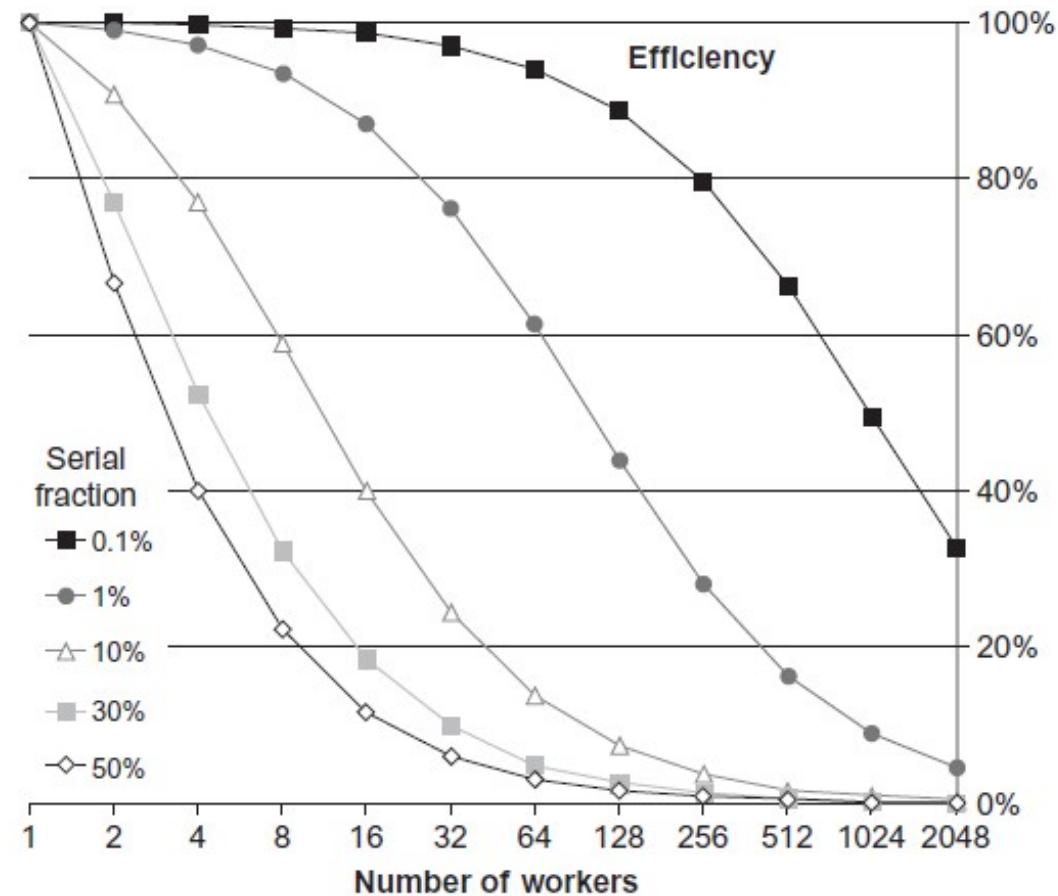
Lei de Amdahl



- A escalabilidade da paralelização é limitada pela fração serial do código. Esta fração é a parte não paralelizada.

PACHECO, Peter S., An introduction to parallel programming

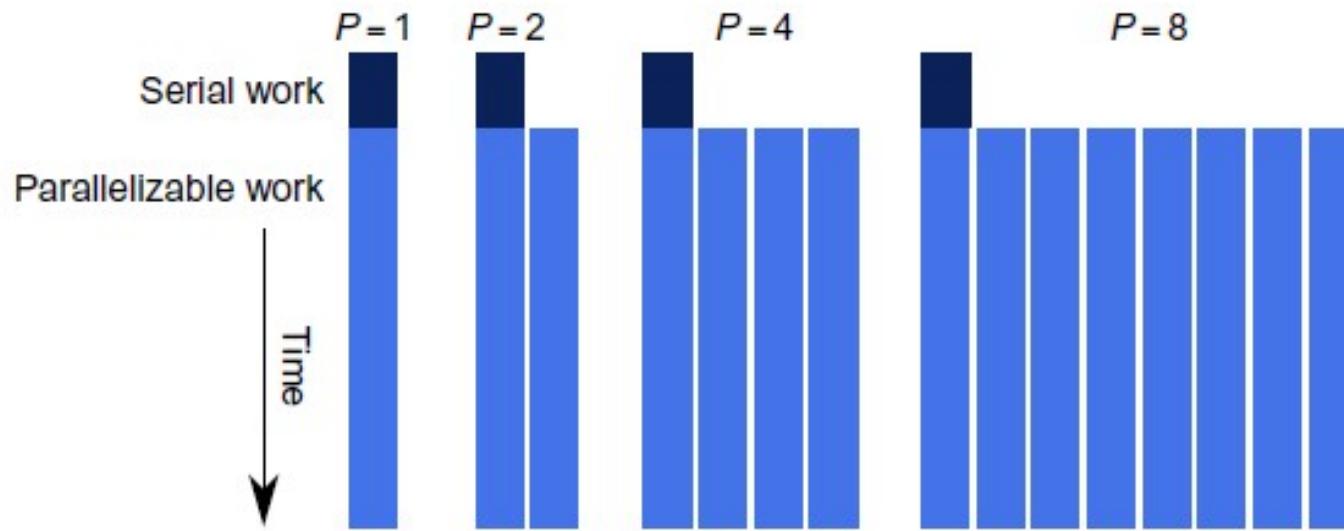
Lei de Amdahl



- Mesmo quando ganhos são possíveis, a **eficiência** pode facilmente ser ruim em função do trecho não paralelizado.

PACHECO, Peter S., An introduction to parallel programming

Lei de Gustafson-Barsis'



- Se o tamanho do problema aumenta em P enquanto o trecho serial cresce lentamente ou se mantém fixo, o ganho cresce se os nós de processamento aumentam em quantidade.

PACHECO, Peter S., An introduction to parallel programming

Strong and Weak Scalability (escalabilidade forte e fraca)

- **Strong scalability:** uma forma de escalabilidade que mede como o desempenho aumenta quando se faz uso de nós adicionais, mas com o tamanho do problema fixo.
 - A lei de Amdahl considera o ganho variando número de nós e problema fixo.
- **Weak scalability:** uma forma de escalabilidade que mede como o desempenho aumenta quando se faz uso de nós adicionais ao mesmo tempo que se aumenta o tamanho do problema, em taxas iguais.
 - A lei de Gustafson-Barsis' Law assume que o tamanho do problema cresce conforme aumenta-se o número de nós.

PACHECO, Peter S., An introduction to parallel programming

Eficiência e Ganho

- Duas importantes métricas relacionadas ao desempenho e paralelismo são ganho e eficiência. O **ganho** compara o tempo para resolver um problema computacional idêntico em uma unidade de processamento versus P unidades:

$$\text{speedup} = S_P = \frac{T_1}{T_P}$$

- onde T1 é o tempo de um programa com uma unidade de processamento e TP é o tempo em P unidades. **Eficiência** é o ganho dividido pelo número de unidades de processamento utilizadas:

$$\text{efficiency} = \frac{S_P}{P} = \frac{T_1}{PT_P}$$

PACHECO, Peter S., An introduction
to parallel programming

Ganho Linear Superlinear

- Um algoritmo que executa P vezes mais rápido em P processadores possui **ganho linear**.
- **Ganho Superlinear:** uma eficiência maior do que 100%.
 - Como isso é possível?
 - Utilização de cache?

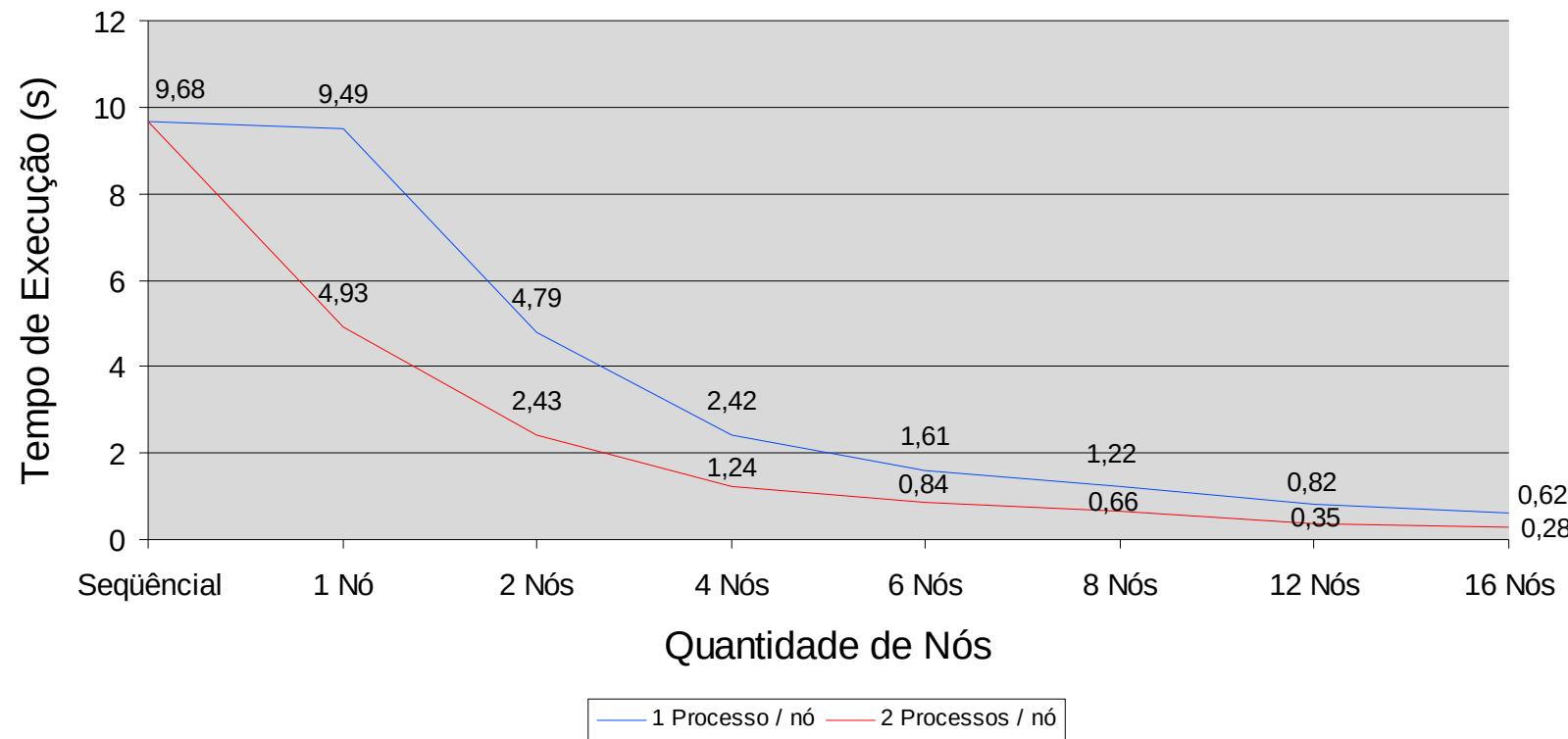
PACHECO, Peter S., An introduction
to parallel programming

Desempenho de uma aplicação paralela

- O que pode impactar o desempenho de uma aplicação paralela reduzindo sua escalabilidade?
 - Rede. Por que?
 - Carga de trabalho desbalanceada. Por que?
 - Regiões sequencias de códigos paralelos. Por que?

Avaliação de desempenho

Resultados de Desempenho

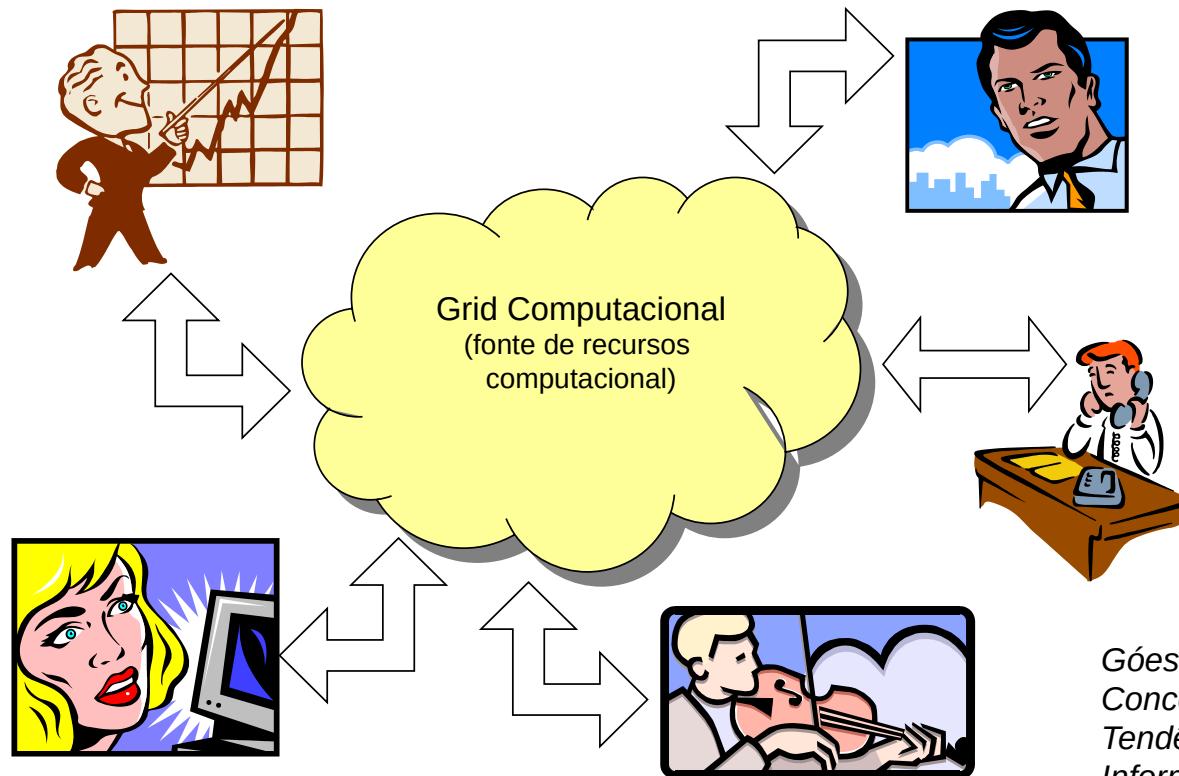


Grid Computacional

- Uma plataforma para execução de aplicações paralelas
 - Amplamente distribuída
 - Heterogênea
 - Compartilhada
 - Sem controle central
 - Com múltiplos domínios administrativos
- Diferença para computação em nuvem (*cloud computing*)?
- O que é *fog computing*?

Grid Computacional

- Analogia com rede elétrica



Góes et al., *Computação em Grade: Conceitos, Tecnologias, Aplicações e Tendências*, Escola Regional de Informática de Minas Gerais (ERI-MG), 2005

Grid Computacional

- SMPs ↑ acoplamento
- MPPs
- NOWs
- Grids ↓ distribuição

- SMP: Symmetric Multiprocessor (memória compartilhada)
- MPP: Massively Parallel Processors
- NOW: Network of Workstations

Grid Computacional

- TeraGrid (de 2001 a 2011)
 - 4 centros de supercomputação norte-americanos
 - Cada centro com milhares de processadores dedicados ao TeraGrid
 - Canais de altíssima velocidade (40 GBits/s)
 - Poder agregado de 13,6 TeraFlops
- SETI@home (desde 1999)
 - Ciclos ociosos de 1.6 milhões de processadores espalhados em 224 países
 - Computa a uma taxa superior a 10 Teraflops
- Grid5000 (desde 2003)
 - Instrumento científico para estudo de sistemas paralelos e distribuídos de larga escala.
 - O objetivo inicial era alcançar 5000 processadores, atualizado para núcleos e alcançado no inverno de 2008-2009.
 - São 9 sites na França (Grenoble, Lille, Luxembourg, Lyon, Nancy, Nantes, Rennes, Sophia-Antipolis, Toulouse).