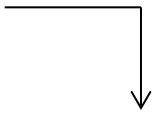
Programação Paralela e Distribuída

Referência:



Motivação:

Necessidade de mais capacidade de cálculo

- Há uma constante procura de maior velocidade de cálculo.
- Problemas de modelação numérica e simulação nas áreas da ciência e engenharia requerem grande capacidade de cálculo.
- Problemas cuja computação deve ser concluída num tempo razoável.

Problemas que representam grandes desafios:

Aplicações tradicionais:

- Modelação de grandes estruturas de ADN.
- Previsão meteorológica.
- Modelação do movimento dos planetas.
- Modelação de movimentos das placas tectónicas.
- Engenharia aeronáutica.
- …

Problemas que representam grandes desafios:

Aplicações mais recentes:

- Motores de busca na web.
- Modelação económica e financeira.
- Realidade Virtual (e.g. Jogos).
- “Big data” e mineração de dados.
- Business Intelligence.
- …

Computação paralela

- Usar mais do que um computador, ou um computador com mais do que processador para resolver um problema.

Porquê

Geralmente para obter computação mais rápida, - ideia base – n processadores a operar simultaneamente serão n vezes mais rápidos !!

- Outro motivos: tolerância a falhas, ter acesso a mais memória, aproveitar recursos disponíveis…

Background

- A ideia de programação paralela e de ter computadores com mais do que um processador existe há mais de **60** anos:

Gill writes in 1958:

"... There is therefore nothing new in the idea of parallel programming, but its application to computers. The author cannot believe that there will be any insuperable difficulty in extending it to computers. It is not to be expected that the necessary programming techniques will be worked out overnight. Much experimenting remains to be done. After all, the techniques that are commonly used in programming today were only won at the cost of considerable toil several years ago. In fact the advent of parallel programming may do something to revive the pioneering spirit in programming which seems at the present to be degenerating into a rather dull and routine occupation ..."

Gill, S. (1958), "Parallel Programming," *The Computer Journal*, vol. 1, April, pp. 2-10.

Conceitos e terminologia

- Supercomputing / high Performance Computing (**HPC**)

- Usar os computadores maiores e mais rápidos para resolver problemas de grande dimensão.

- **Node** – um computador, geralmente possui vários processadores, memória, interfaces de rede.

- **CPU/** processor/ **core**

- Eventualmente podemos ter um nó com múltiplos CPUs e cada CPU com múltiplos cores (ou núcleos), sendo cada um, uma unidade de execução.

Conceitos e terminologia

- **Task** – Secção de código que constitui uma tarefa a executar.

- **Shared Memory**

Em termos de hardware descreve uma arquitetura em que todos os processadores têm acesso a um área de memória física comum.

Em termos de programação descreve um modelo em que as várias tarefas paralelas têm acesso a uma memória lógica comum.

Conceitos e terminologia

- **Symmetric Multi-Processor (SMP)**

Arquitetura (hardware) de memória partilhada em que vários processadores partilham o mesmo espaço de endereçamento e têm acesso igual a todos os recursos**.**

- **Distributed Memory**

Em termos de hardware descreve uma arquitetura em que os vários processadores não partilham a memória física

Em termos de programação descreve um modelo em que as várias tarefas paralelas não partilham memória, têm de comunicar por mensagens.

Conceitos e terminologia

- **Synchronization -** coordenação de tarefas

- **Granularity –** medida qualitativa do rácio entre computação e comunicação.

Coarse – Grande quantidade de cálculo realizado entre cada operação de comunicação.

Fine – poucos cálculos entre operações de comunicação.

Speedup Factor

$$S(p) = \frac{\text{Execution time using one processor (best sequential algorithm)}}{\text{Execution time using a multiprocessor with } p \text{ processors}} \quad \frac{t_s}{t_p}$$

Onde t_s é o tempo de execução num único processador e t_p é o tempo de execução num multiprocessador.

$S(p)$ dá o ganho em tempo de execução por usar vários processadores.

Deve ser usado o algoritmo sequencial mais rápido.

O Speedup pode ser calculado em termos de número de instruções:

$$S(p) = \frac{\text{Number of computational steps using one processor}}{\text{Number of parallel computational steps with } p \text{ processors}}$$

Pode também ser estendido à noção de complexidade temporal.

Eficiência

$$E = \frac{Tempo\ de\ execução\ sequencial}{Tempo\ de\ execução\ num\ multiprocessador \times número\ de\ processadores} = \frac{t_s}{t_p \times p}$$

Em percentagem,

$$E = \frac{S(P)}{p} \times 100\%$$

Eficiência é 100% quando o speedup é p.

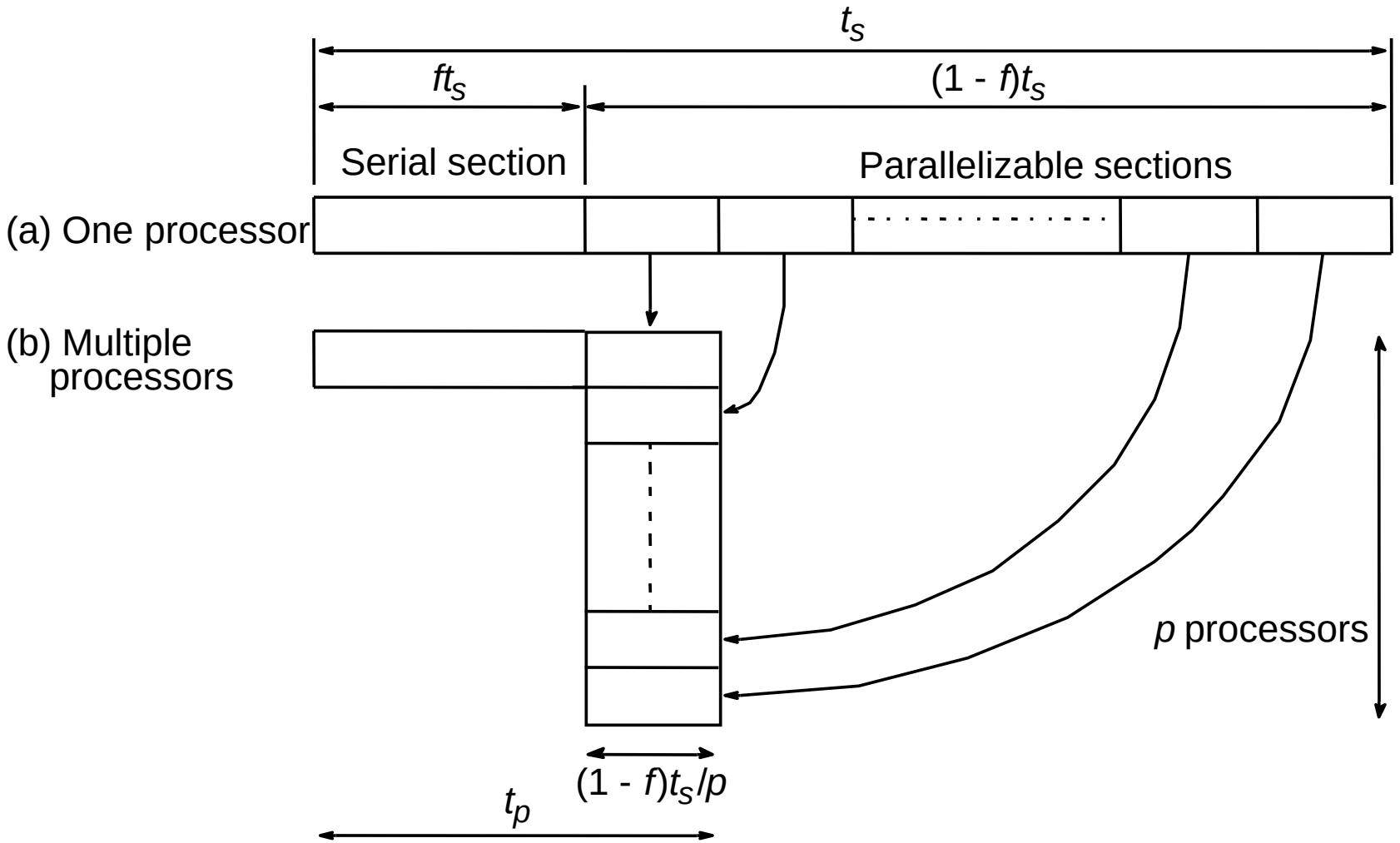
Speedup Máximo

Geralmente o speedup máximo é *p* com *p* processadores (speedup linear).

É possível obter um speedup super linear (maior que *p*) mas em situações particulares:

- Acesso a mais memória na versão paralela.
- Algoritmos não determinísticos.
- …

Speedup Máximo
Lei de Amdahl



Equação conhecida como lei de Amdahl

O Speedup é dado por:

$$S(p) = \frac{t_s}{ft_s + (1-f)t_s/p} = \frac{p}{1+(p-1)f}$$

$$\text{Speedup} = p / (1+(p - 1)f)$$

Se nada pode ser paralelizado, todo o programa é sequencial:

f = 1 (100%)

Processadores (p)	Speedup
2	1
3	?
4	?

$$\text{Speedup} = p / (1+(p - 1)f)$$

Se 50% é sequencial:

f = 0.5

Processadores (p)	Speedup
2	1,33
3	1,5
4	??
100 000	1,99

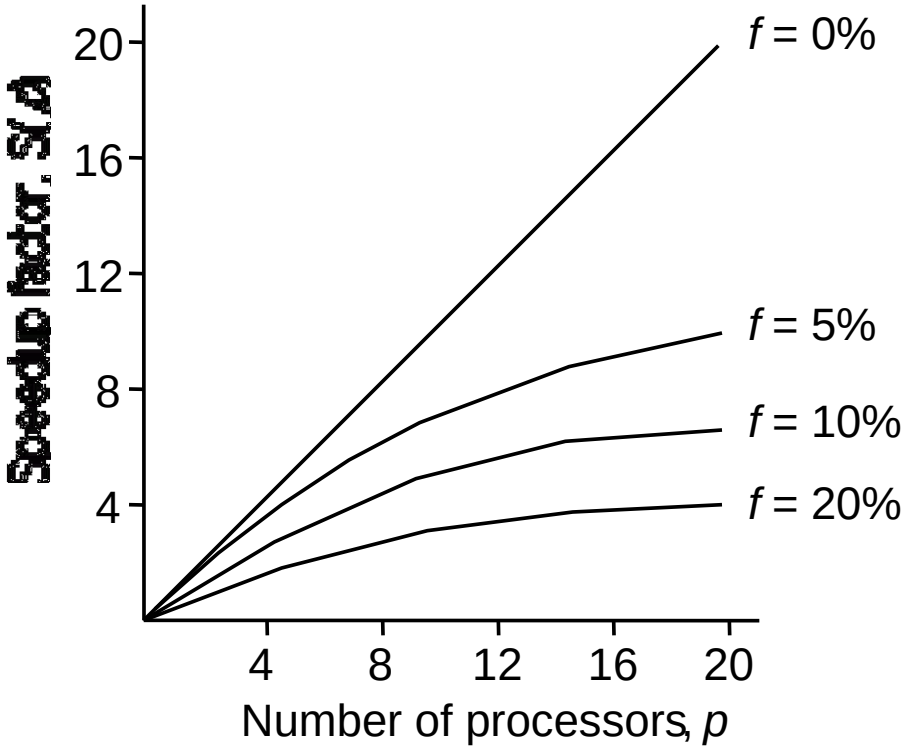
$$\text{Speedup} = p / (1+(p - 1)f)$$

Se 0% é sequencial, todo o programa é paralelizável:

f = 0.0

Processadores (p)	Speedup
2	?
3	?
4	?
100 000	?

Speedup against number of processors



Mesmo com um número infinito de processadores o máximo speedup é limitado a 1/*f*.

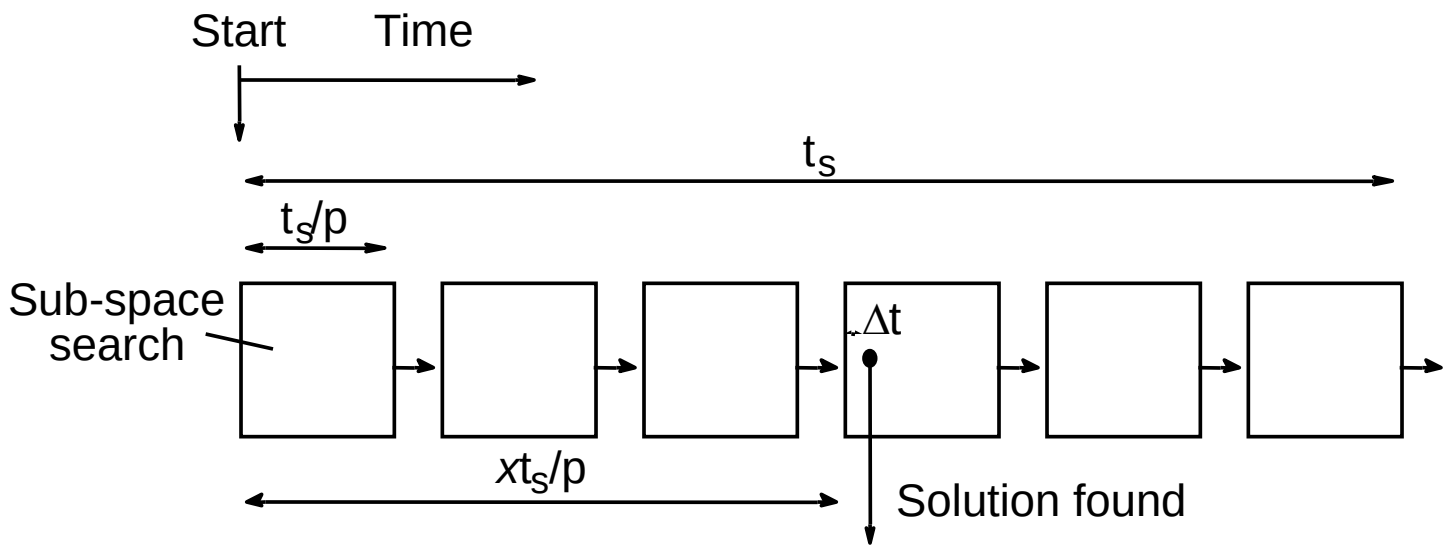
Exemplo

Segundo a lei de Amdahl, mesmo que apenas 5% da computação seja sequencial o máximo speedup é 20, independentemente do número de processadores.

!!!

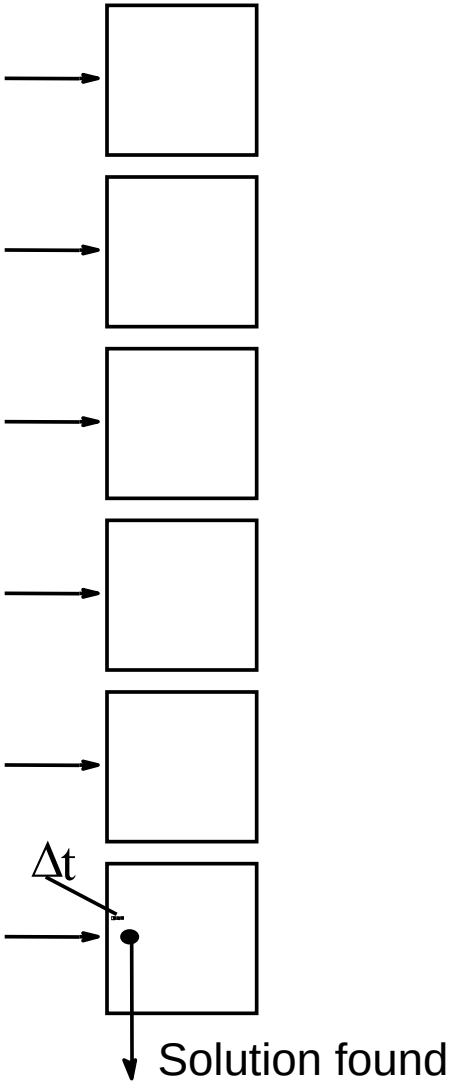
Exemplo de Speedup super linear - Pesquisa

(a) Pesquisar sequencialmente cada sub-espaço



x indeterminate

(b) Pesquisar cada sub-espaço em paralelo



Speed-up é dado por:

$$S(p) = \frac{\left\lceil x \times \frac{t_s}{p} \right\rceil + \Delta t}{\Delta t}$$

Pesquisa sequencial: pior caso quando a solução é encontrada no último sub-espaço.

Versão paralela é vantajosa.

$$S(p) = \frac{\left\lceil \frac{p-1}{p} \right\rceil \times t_s + \Delta t}{\Delta t} \rightarrow \infty$$

Quando Δt tende para zero

A menor vantagem para a versão paralela ocorre quando a solução é encontrada no primeiro sub-espaço

$$S(p) = \frac{\Delta t}{\Delta t} = 1$$

Escalabilidade

O desempenho de um sistema depende do número de processadores:

O sistema é escalável se aumentando o número de processadores, a resolução de problemas de igual dimensão é mais rápida;

Escalabilidade

O desempenho de um sistema depende do algoritmo usado:

O sistema é escalável se com o mesmo número de processadores, é possível resolver problemas de maior dimensão no mesmo tempo.

Na prática, quando os recursos computacionais aumentam, a dimensão dos problemas, em particular a dimensão do conjunto de dados de input, também pode aumentar.

Geralmente , a dimensão da componente paralelizável cresce mais rápido que a componente sequencial.

Exemplo: produto de matrizes.

Tipos de Computadores Paralelos

Dois tipos principais:

- Memória partilhada

(Shared memory multiprocessor)

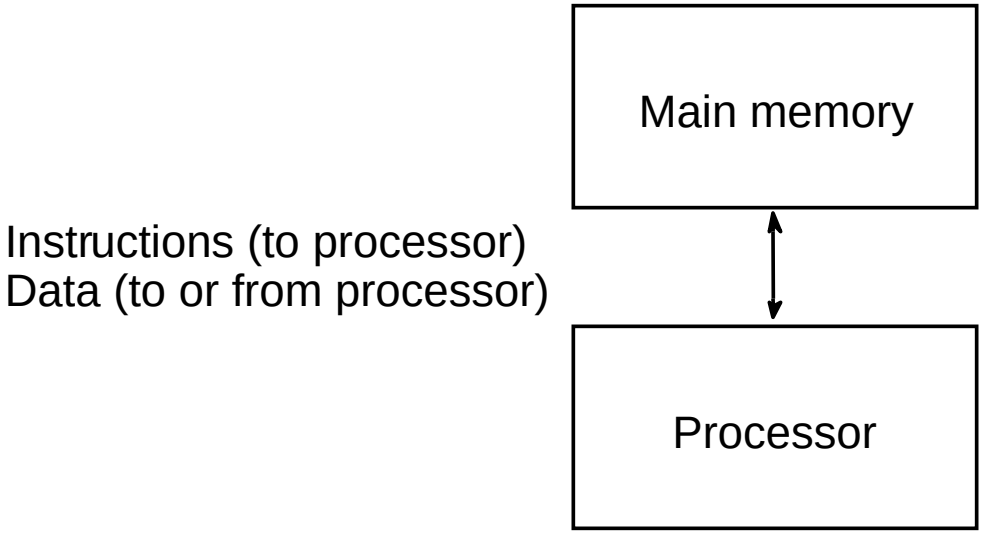
- Memória distribuída

(Distributed memory multicomputer)

Shared Memory Multiprocessor

Computador convencional

Consiste em um processador que executa um programa armazenado na memória principal:

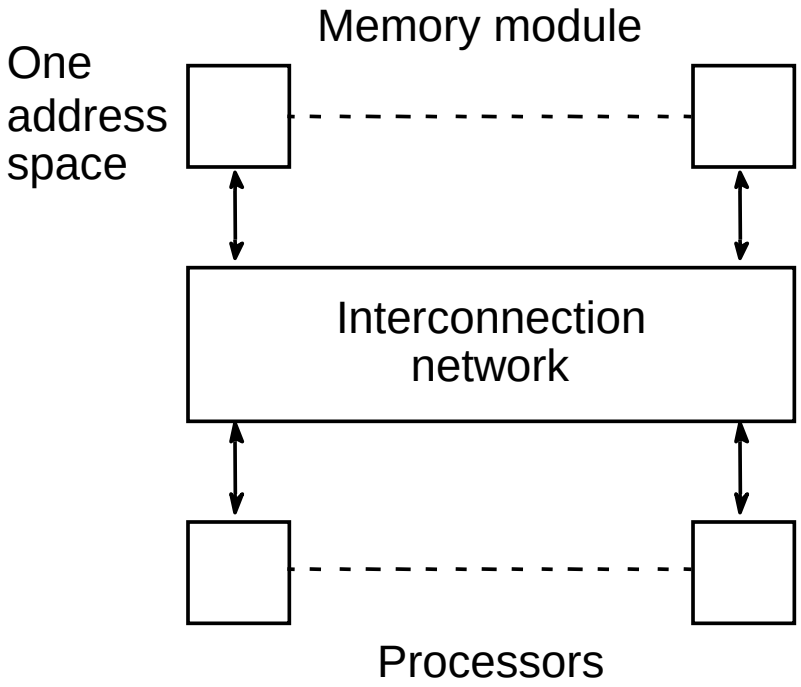


Cada posição de memória é localizada pelo seu endereço.

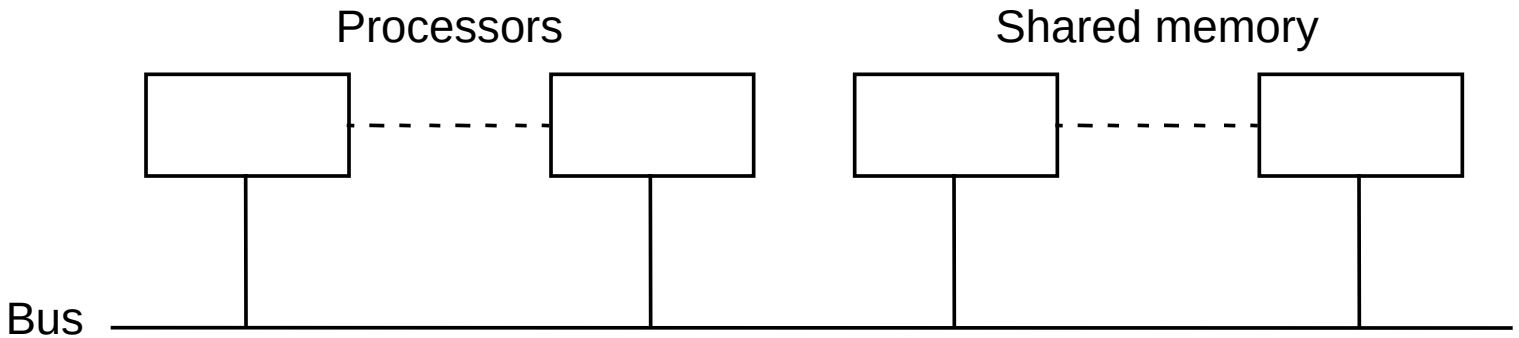
Os endereços começam em 0 e vão até $2^b - 1$, com e *b* o número de bits (binary digits) disponíveis.

Shared Memory Multiprocessor System

Múltiplos processadores ligados a vários módulos de memória, tal que cada processador pode aceder a qualquer módulo de memória:



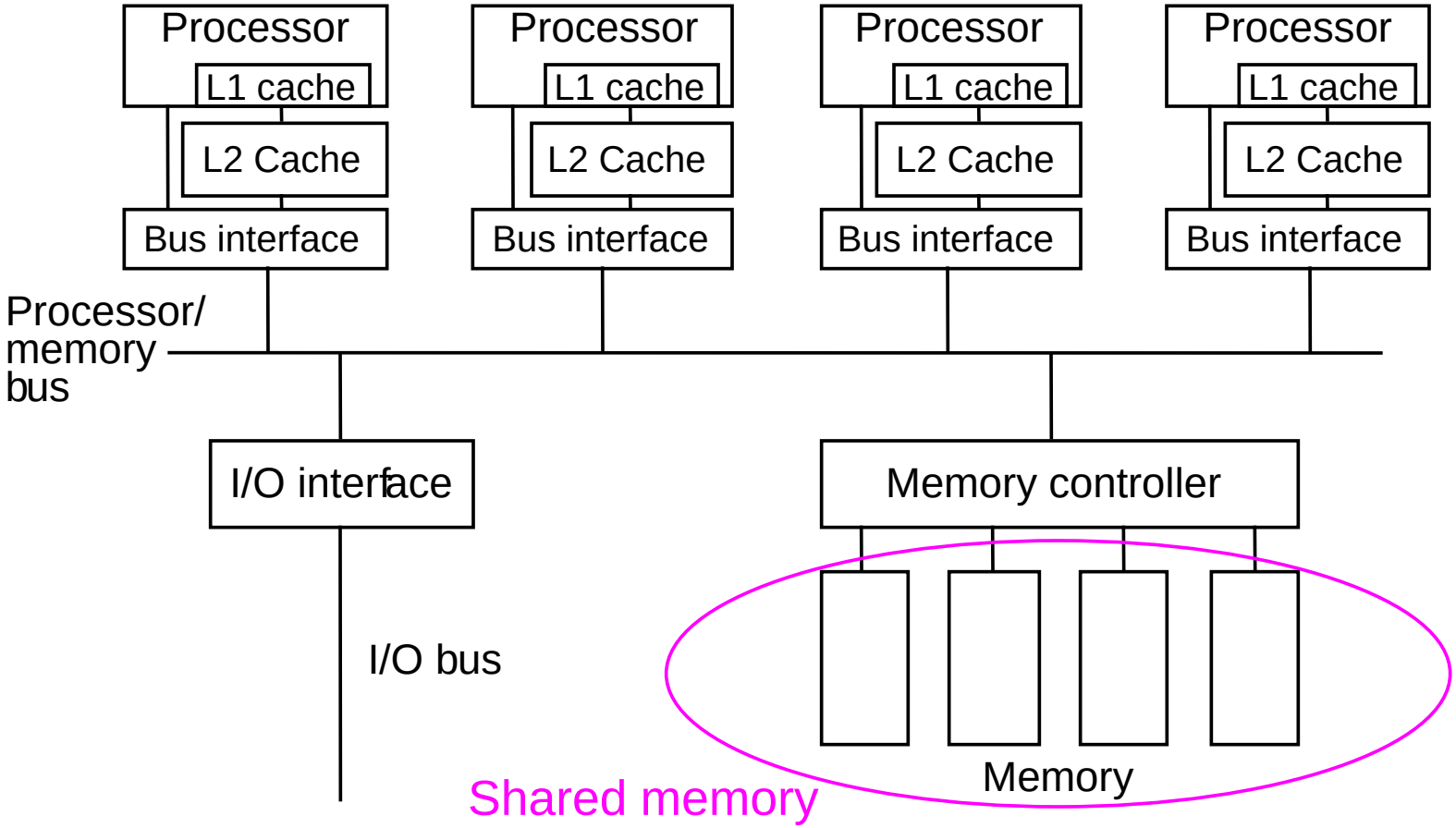
Visão simplista de um multiprocessador de memória partilhada:



Examples:

- Dual Pentiums
- Quad Pentiums

Quad Pentium Shared Memory Multiprocessor



Vantagens / desvantagens de multiprocessadores de memória partilhada

+ Interface de programação fácil.
+ Patilha de dados entre tarefas rápida e uniforme.

- Comunicação entre memória e CPU não escalável. Adicionar mais CPUs, aumenta o tráfego entre CPUs e memória.
- Programador é responsável por sincronizar acesso a dados partilhados.

Programação de Multiprocessadores de memória partilhada

Modelos de memória partilhada (sem threads):

- Vários processos / tarefas partilham um espaço de endereçamento comum.
- Mecanismos como locks e semáforos permitem sincronizar os acessos.

Exemplos:

POSIX, Unix

- fornecem funções para criar segmentos de memória partilhada que podem ser acedidos por todos os processos que lhe são associados.

Programação de Multiprocessadores de memória partilhada

- Threads – Um processo pode criar várias sequências de execução paralela (threads), cada uma podendo aceder às variáveis globais do processo.

Exemplos:

POSIX threads (Pthreads);

Java threads

Python threads

Cuda threads (para GPU)

…

- Linguagem de programação sequencial com um pre-processador para directivas que declaram variáveis partilhadas e especificam paralelismo.

Exemplo:

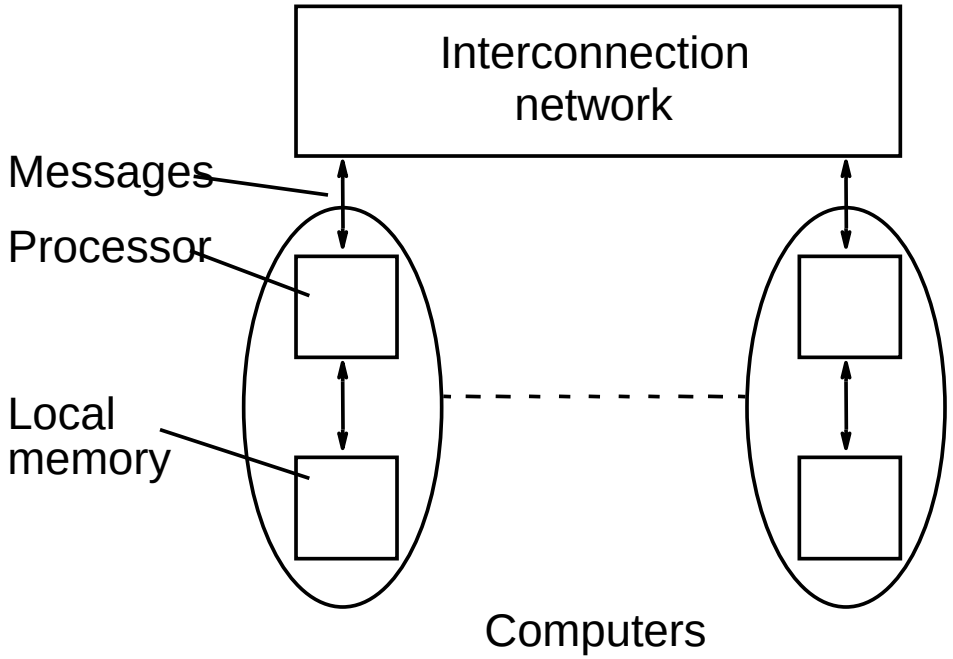
OpenMP (Open Multi-processing)

– standard industrial

Distributed Memory Multicomputer (Message passing model)

Message-Passing Multicomputer

Vários computadores ligados por uma rede de comunicação



Message-Passing Multicomputer

Vários computadores ligados por uma rede de comunicação.

- Um conjunto de tarefas que usam a sua memória local na computação; Podem existir várias tarefas na mesma máquina física ou distribuídas por diferentes máquinas.

- As tarefas trocam dados através de mensagens.

Message-Passing Multicomputer

- Transferência de dados entre tarefas exige:
- cooperação entre processos (uma operação de envio (send) exige uma operação de aceitação (receive),

ou

- Um sistema de gestão de mensagens.

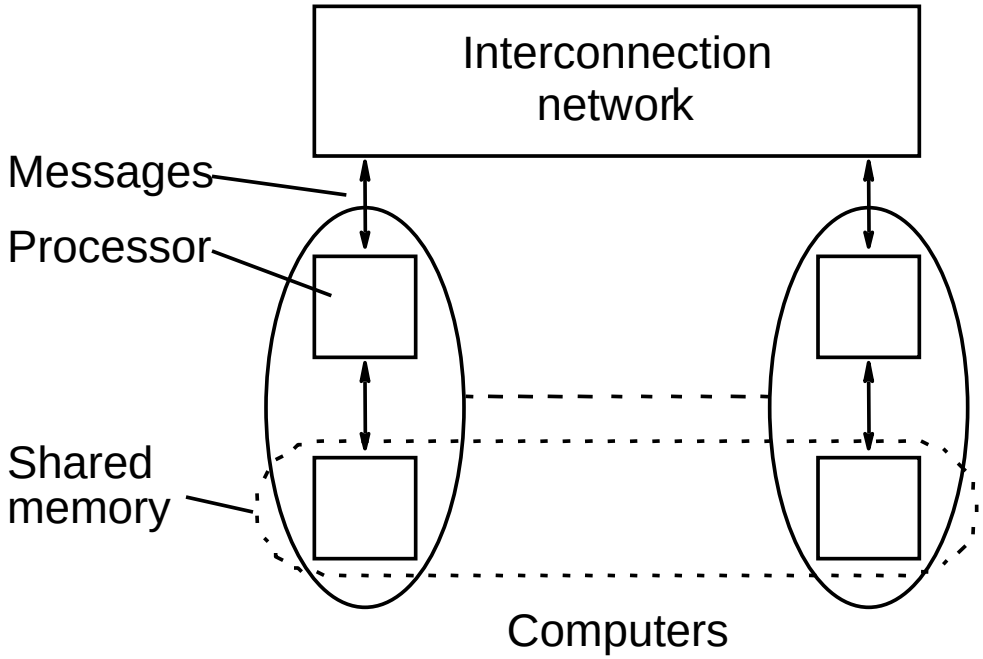
Programação do modelo de comunicação por mensagens

MPI – Message Passing Interface

- Standard industrial para comunicação por mensagens.

Distributed Shared Memory

Fazer a memória principal de um grupo de computadores interligados funcionar como um único espaço de memória. Podem depois usar-se as técnicas de programação com memória partilhada.



Classificação de Flynn

Flynn (1966) criou uma classificação baseada na sequência de instruções e de dados:

- *Single instruction stream-single data stream (SISD) computer*

Computador com um único processador. Um único fluxo de instruções opera num único fluxo de dados.

Multiple Instruction Stream-Multiple Data Stream (MIMD) Computer

Multiprocessador de uso geral (general-purpose)

- Cada processador executa um fluxo de execução diferente e cada instrução pode operar em dados diferentes.

Quer o modelo de memória partilhada quer o modelo de comunicação por mensagens podem seguir este modelo.

Single Instruction Stream-Multiple Data Stream (SIMD) Computer

- Um único fluxo de execução opera em diferentes fluxo de dados.

Cada processador executa a mesma instrução em diferentes conjuntos de dados.

Exemplo:

Graphical processing units (GPU)

Multiple Instruction Stream-Single Data Stream (MISD) Computer

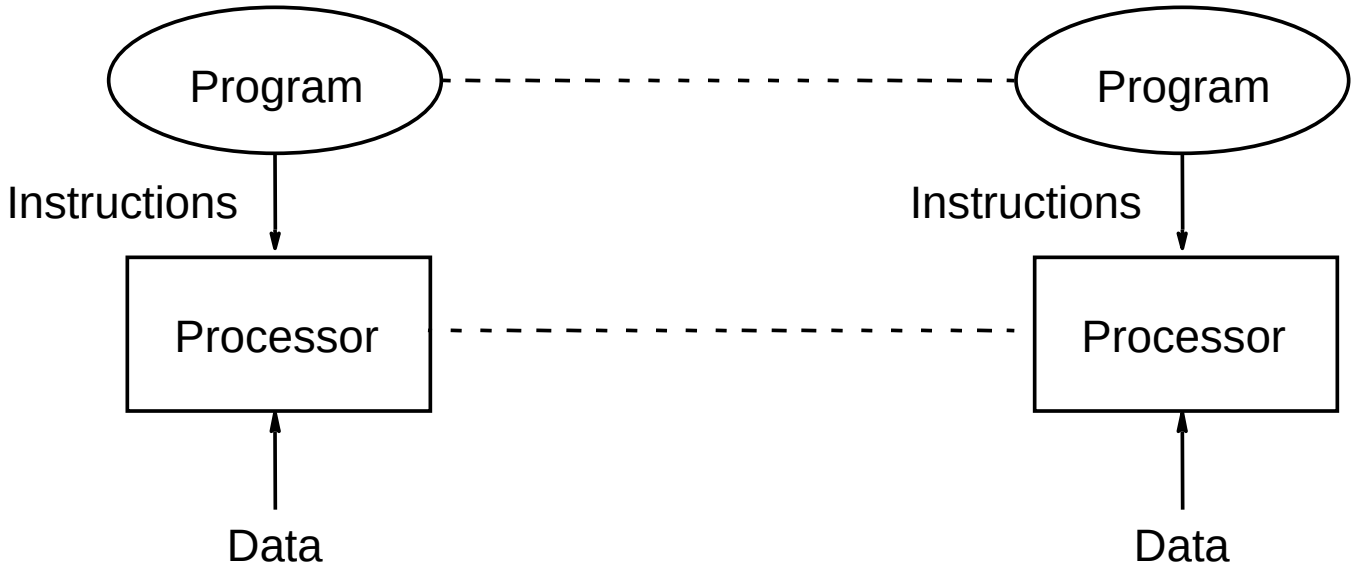
- Múltiplas instruções operam no mesmo fluxo de dados.

Usado por exemplo para tolerância a falhas. Vários sistemas heterogéneos operam sobre os mesmos dados e os resultados são comparados.

Modelos de programação paralela mais comuns

Multiple **Program** Multiple Data (MPMD) Structure

Dentro da classificação MPMD, cada processador executa o seu programa:



Single **Program** Multiple Data (SPMD) Structure

Existe um único código fonte e cada processador executa a sua cópia do programa de forma independente.

O programa pode ser construído de forma a que partes do programa são executadas e outras não, dependendo do identificador do computador.

Networked Computers as a Computing Platform

- Usar um conjunto de computadores comuns, ligados em rede, é uma alternativa barata ao uso de supercomputadores.

- Alguns dos primeiros projetos:

  - Berkeley NOW (network of workstations) project.
  - NASA Beowulf project.

Vantagens:

- Conjunto de PCs disponíveis a baixo custo
- Os processadores mais recentes podem ser incorporados no sistema à medida que ficam disponíveis.

Software para Clusters

- Baseado em comunicação por mensagens:
- Parallel Virtual Machine (PVM) – finais dos anos 80.
- Message-Passing Interface (MPI) - standard definido no anos 90.
- Ambos fornecem um conjunto de bibliotecas de comunicação por mensagens. Podem ser usadas com linguagens de programação comuns: (C, C++, ...)

Super-computadores atuais

- https://www.top500.org/ ➔

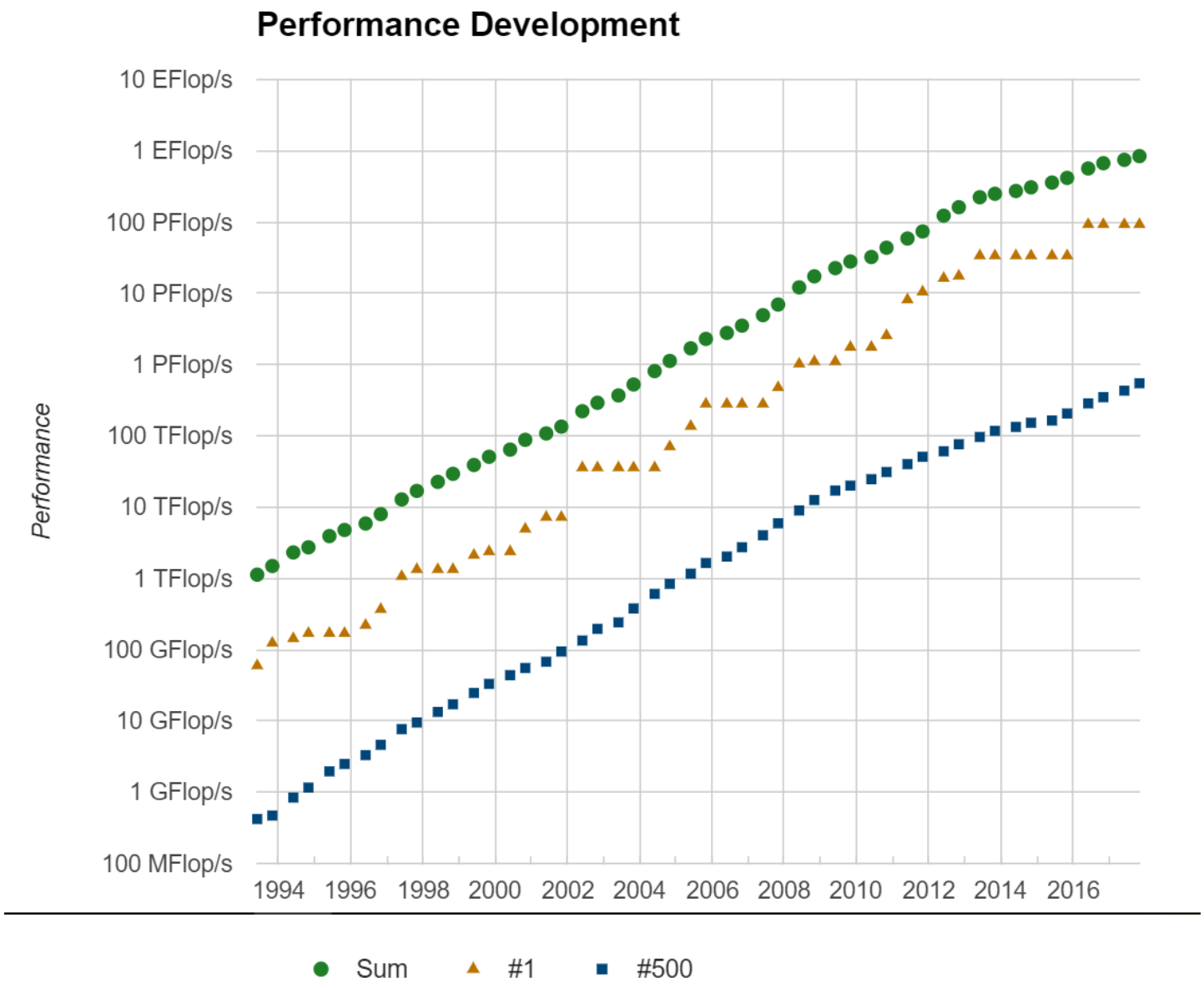
Rmax *- maximal achieved performance*

Rpeak *- theoretical peak performance*

Valores obtidos com o LINPACK Benchmark.

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
4	Japan Agency for Marine-Earth Science and Technology Japan	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz ExaScaler	19,860,000	19,135.8	28,192.0	1,350
5	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209

		Sugon				
495	Trading Company China	**DLSystem** - Sugon W580-G20, Xeon E5-2640v3 8C 2.6GHz, 10G Ethernet, NVIDIA Tesla K80 Sugon	13,440	551.1	912.3	168
496	Telecom Company Mexico	Cluster Platform DL380, Xeon E5-2680v4 14C 2.4GHz, Infiniband FDR HPE	18,928	550.7	726.8	
497	Telecom Company China	Lenovo NeXtScale nx360M5, Intel Xeon E5-2660v2 10C 2.2GHz, 10G Ethernet Lenovo	78,000	550.1	1,372.8	3,510
498	WETA Digital New Zealand	Apollo 6000 XL230a, Xeon E5-2680v3 12C 2.5GHz, 10G Ethernet HPE	21,600	549.1	864.0	
499	NASA/Goddard Space Flight Center United States	**Discover SCU12** - Rackable Cluster, Xeon E5-2697v3 14C 2.6GHz, Infiniband FDR HPE	17,136	548.7	712.9	1,224
500	NASA/Goddard Space Flight Center United States	**Discover SCU11** - Rackable Cluster, Xeon E5-2697v3 14C 2.6GHz, Infiniband FDR HPE	17,136	548.7	712.9	1,224



Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}