TN741 - Computação de Alto Desempenho

Introdução

Marcelo Zamith

e-mail: mzamith@ufrrj.br

https://www.dcc.ufrrj.br/~marcelo/

Universidade Federal Rural do Rio de Janeiro - DCC

Introdução

Introdução

- A computação de alto desempenho:

Introdução

- A computação de alto desempenho:
  - HPC → ***H**igh **P**erformance **C**omputing*

Introdução

- A computação de alto desempenho:
  - HPC → ***H**igh **P**erformance **C**omputing*
- O que é computação de alto desempenho ?
- Para que serve a computação de alto desempenho ?

Introdução

- Algumas frases para refletir:

Introdução

- Algumas frases para refletir:
  1. “Quatro ou cinco computadores devem ser suficientes para o mundo inteiro até o ano 2000.”

Introdução

- Algumas frases para refletir:

  1. “Quatro ou cinco computadores devem ser suficientes para o mundo inteiro até o ano 2000.”

     T.J. Watson, Presidente da IBM, 1945.

Introdução

- Algumas frases para refletir:
  1. “Quatro ou cinco computadores devem ser suficientes para o mundo inteiro até o ano 2000.”

     T.J. Watson, Presidente da IBM, 1945.
  2. “640KB [de memória] deve ser suficiente para qualquer um.”

Introdução

- Algumas frases para refletir:
  1. “Quatro ou cinco computadores devem ser suficientes para o mundo inteiro até o ano 2000.” T.J. Watson, Presidente da IBM, 1945.
  2. “640KB [de memória] deve ser suficiente para qualquer um.” Bill Gates, Presidente da Microsoft, 1981.

Introdução

A Google:

A computação de alto desempenho (HPC) é a prática de **agregar recursos** de computação para conseguir um **desempenho maio**r do que o de **uma única estação de trabalho, servidor ou computador**. A HPC pode assumir a forma de **supercomputadores personalizados** ou **grupos de computadores individuais** chamados **clusters**. A HPC pode ser executada no local, na nuvem ou como um híbrido de ambos. Cada computador em um ***cluster*** é geralmente chamado de nó, e cada nó é responsável por uma tarefa diferente.[1]

Introdução

A IBM:

A computação de alto desempenho (HPC) é uma tecnologia que usa ***clusters*** de processadores potentes que trabalham em **paralelo** para processar **conjuntos de dados multidimensionais maciços**, também conhecidos como ***big data***, e resolver problemas complexos em **velocidades extremamente altas**. A HPC resolve alguns dos **problemas** de computação mais **complexos** da atualidade em **tempo real**.[2]

Introdução

A Ciência

- Formular uma teoria e escrever um algoritmo.
- Realizar experimentos experimentos e construir sistemas:
  - Muito difícil - construir grandes túneis de vento.
  - Demasiado caro - construir um avião de jogar fora.
  - Demasiado lento - espere clima ou evolução galáctica.
  - Demasiado perigoso - armas, projeto da droga, experiências do clima

Introdução

Para que serve a computação de alto desempenho ?

- Ciência
  - Modelagem climática global e modelagem de furacões
  - Modelagem Astrofísica
  - Biologia: genômica; dobramento de proteínas; Projeto de drogas
  - Química computacional
  - Ciências dos Materiais Computacionais e Nanociências
- Engenharia
  - Simulações físicas (tempo, desastres, etc...)
- Negócio
  - Modelagem financeira e econômica
  - Processamento de transações, serviços web e motores de busca

Introdução

- Computadores vistos como máquinas seqüências.
- Programas e algoritmos definidos como uma seqüência de passos.
- Execução do programa em uma seqüência de instruções, uma instrução por vez.
- Instruções:
  - Execução de microinstruções e sinais simultaneamente.
  - Uso de *pipelines* para execução de instruções.

Introdução

- Paralelismo de instrução super escalar.
- Várias unidades de execução dentro de um único processador.
- Execução de múltiplas instruções do mesmo programa em paralelo.
- Baixo custo de *hardware*.
- Melhora de desempenho.
- *Stream multiprocessors* (SMs).
- Compartilhamento de memória.
- Coerência de cache.

Introdução

- Paralelismo de instrução super escalar.
- Várias unidades de execução dentro de um único processador.
- Execução de múltiplas instruções do mesmo programa em paralelo.
- Baixo custo de *hardware*.
- Melhora de desempenho.
- *Stream multiprocessors* (SMs).
- Compartilhamento de memória.
- Coerência de cache.

Introdução

Supercomputador brasileiro - **Santos Dumont**

Introdução

Supercomputador brasileiro - **Santos Dumont**

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM
  - 246 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 384Gb de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM
  - 246 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 384Gb de memória RAM
  - 36 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 768 Gb de memória RAM

Introdução

Supercomputador brasileiro

- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM
  - 246 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 384Gb de memória RAM
  - 36 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 768 Gb de memória RAM
  - 94 nós computacionais (GPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252, 4x NVIDIA Volta V100 GPU e 384Gb de memória RAM

Introdução

Supercomputador brasileiro

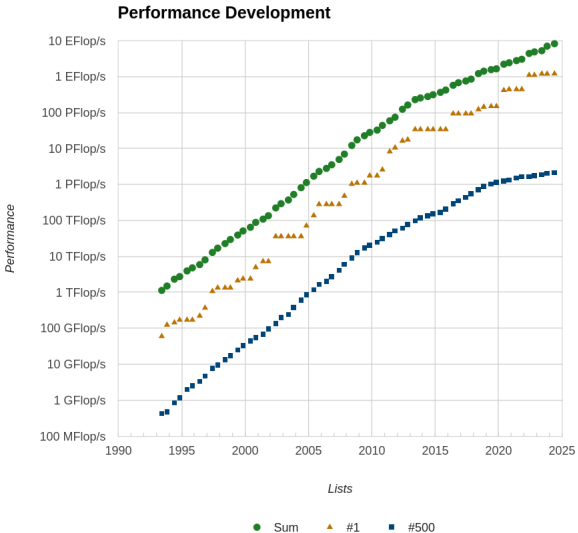
- **Santos Dumont**: Capacidade para 5,1 Petaflop/s
  - 504 nós de computação B710 (thin node), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 198 nós de computação B715 (thin node) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 54 nós de computação B715 (thin node) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
  - 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM
  - 246 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 384Gb de memória RAM
  - 36 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 768 Gb de memória RAM
  - 94 nós computacionais (GPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252, 4x NVIDIA Volta V100 GPU e 384Gb de memória RAM
  - 1 nó para Inteligência Artificial com 2x Intel Xeon Skylake Gold 6148 (20c @2,4GHz), 8x NVIDIA Tesla V100-16GB com NVLink e 384Gb de memória RAM

Introdução

$$1\ petaFLOP = 1000000000000000 = 1 \times 10^{15}$$

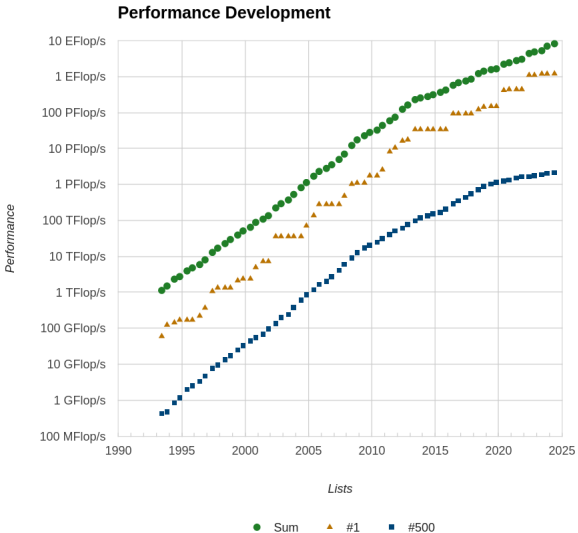
Nome	Unidade	Valor
kiloFLOPS	KFLOPS	10^3
megaFLOPS	MFLOPS	10^6
megaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}

Introdução - Top 500 *Super Computing*



Fonte: `https://www.top500.org/statistics/perfdevel/` - acessado em 03/08/2024.

Introdução - Top 500 *Super Computing*



- Santos Dumont 5.1$PFLOPS/S$
- iPhone 7 1.5$GFLOPS/S$

Fonte: `https://www.top500.org/statistics/perfdevel/` - acessado em 03/08/2024.

Introdução

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, **HPE** DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,206.00	1,714.81	22,786
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, **Intel** DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, **Microsoft Azure** Microsoft Azure United States	2,073,600	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, **Fujitsu** RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, **HPE** EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107

Fonte: Top 500 *Super Computing* (`https://www.top500.org/`) - acessado em 03/08/2024.

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
UFRRJ

Introdução

Green500 Data

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	189	**JEDI** - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, ParTec/EVIDEN EuroHPC/FZJ Germany	19,584	4.50	67	72.733
2	128	**Isambard-AI phase 1** - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11, HPE University of Bristol United Kingdom	34,272	7.42	117	68.835
3	55	**Helios GPU** - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11, HPE Cyfronet Poland	89,760	19.14	317	66.948
4	328	**Henri** - ThinkSystem SR670 V2, Intel Xeon Platinum 8362 32C 2.8GHz, NVIDIA H100 80GB PCIe, Infiniband HDR, Lenovo Flatiron Institute United States	8,288	2.88	44	65.396
5	71	**preAlps** - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11, HPE Swiss National Supercomputing Centre (CSCS) Switzerland	81,600	15.47	240	64.381

Fonte: *Top Green500 List* (https://www.top500.org/) - acessado em 03/08/2024

Como chegamos aos supercomputadores ?

Introdução

- Super Computador:
  - É/foi considerado o termo que refere-se a capacidade de processamento e velocidade de cálculos.
  - Termo Super Computador - 1929 no jornal New York World - Máquina da IBM para a Universidade de Colúmbia.
  - Originalmente o termo Super Computador foi designado por Seymour Roger Cray (Pai do Super Computador).
  - Em 1970 os super computadores eram empregados em processamento vetorial

Introdução

- Benchmark: LINPACK (Jack Dongarra) - operações de ponto flutuante de 64bits:
  - Mede o tempo para resolver um sistema de equações lineares $n \times n$.
- HPLinpack:
  - Mede o tempo para resolver um sistema de equações lineares $n \times n$ usando decomposição LU - desenvolvidos para computadores paralelos.

Introdução

- Evolução:
  - A lei de Moore (1975):

Introdução

- Evolução:
  - A lei de Moore (1975): Os quantidade transistores dobraria com um baixo custo a cada 2 anos!

Introdução

- Evolução:
  - A lei de Moore (1975): Os quantidade transistores dobraria com um baixo custo a cada 2 anos!
  - CPUs - *Central Processing Unit*
    - Multicores
    - Clusters
  - GPUs - *Graphics Processing Unit*
    - Aceleradoras para processamento genéricos

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz
- 1985: Intel 386DX/32 bits/275.000 transistores/33 MHz

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz
- 1985: Intel 386DX/32 bits/275.000 transistores/33 MHz
- 1995: Intel Pentium Pro/64 bits/5,5 milhões transistores/200 MHz

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz
- 1985: Intel 386DX/32 bits/275.000 transistores/33 MHz
- 1995: Intel Pentium Pro/64 bits/5,5 milhões transistores/200 MHz
- 2000: Intel Pentium 4/64 bits/42 milhões transistores/1.8 GHz ←

Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz
- 1985: Intel 386DX/32 bits/275.000 transistores/33 MHz
- 1995: Intel Pentium Pro/64 bits/5,5 milhões transistores/200 MHz
- 2000: Intel Pentium 4/64 bits/42 milhões transistores/1.8 GHz ←
- 2006: Intel Core 2 Duo/64 bits/167 milhões transistores/1.2 GHz

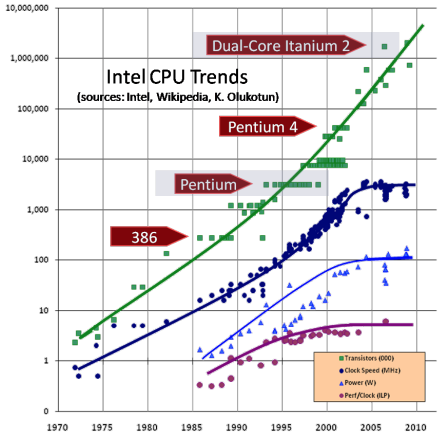
Introdução

CPU

- 1971: Intel 4004/4 bits/2.300 transistores/108 kHz
- 1972: Intel 8008/8 bits/3.500 transistores/108 kHz
- 1978: Intel 8086/16 bits/29.000 transistores/10 MHz
- 1985: Intel 386DX/32 bits/275.000 transistores/33 MHz
- 1995: Intel Pentium Pro/64 bits/5,5 milhões transistores/200 MHz
- 2000: Intel Pentium 4/64 bits/42 milhões transistores/1.8 GHz ←
- 2006: Intel Core 2 Duo/64 bits/167 milhões transistores/1.2 GHz
- 2024: Intel Core i9-14900KS/64 bits/ 24 cores / 6.2 GHz

Introdução

CPU



http://www.gotw.ca/publications/concurrency-ddj.htm. Acessado em 03/08/2024

Introdução

- Máquinas desktop formando pequenos *clusters*
- Aceleradoras massivamente paralelas
- Superpipelines, registradores escalares e instruções vetoriais.
- Dispositivos multiprocessados - super computador de hoje - computador de casa amanhã

Introdução

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
UFRRJ

Introdução

- Sansung S7
  - Processador:
    - Dual-core 2.15 GHz Kryo & dual-core 1.6 GHz Kryo
    - (Exynos) Quad-core 2.3 GHz Cortex-A53 + quad-core 1.6 GHz Cortex-A53
  - Memória:
    - 4 GB de RAM

Introdução

- Sansung S7
  - Processador:
    - Dual-core 2.15 GHz Kryo & dual-core 1.6 GHz Kryo
    - (Exynos) Quad-core 2.3 GHz Cortex-A53 + quad-core 1.6 GHz Cortex-A53
  - Memória:
    - 4 GB de RAM

- $\approx 2.246 \times$ mais rápido em relação ao clock.
- $2.097.152 \times$ mais capacidade de memória

Computação na nuvem

Computação na nuvem

Internet das coisas - IoT

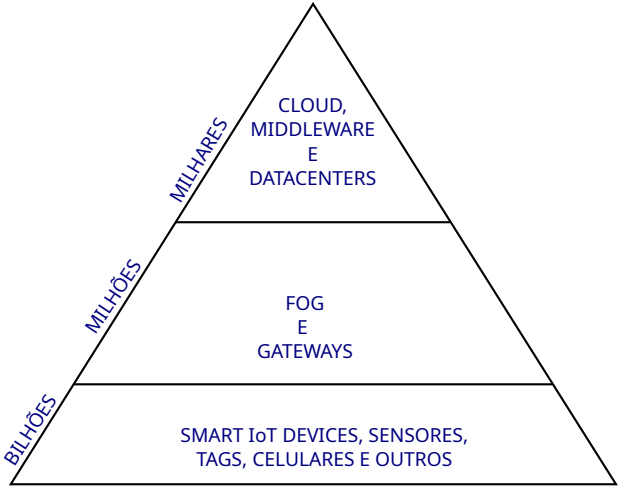
- Evolução de elementos tecnológicos: sensores, hardware, semântica, armazenamento, processamento, comunicação, entre outros
- Unidos em um mesmo ambiente.
- Futuro da computação e das comunicações.

Observação:

IoT busca melhorar o mundo de forma que os objetos tenham autonomia e conhecimento, viabilizando que funcione de forma inteligente sem a necessidade de instruções.

Computação na nuvem

Internet das coisas - IoT



Computação na nuvem

- Poder computacional.
- Disponibilidade.
- Não isolamento dos dados.
- Desperdício de processamento.

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Equipamentos ociosos.
- Custo: manutenção de *hardware* e *software*.
- Alto custo do backup - múltiplos pontos de armazenamento.

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Equipamentos ociosos.
- Custo: manutenção de *hardware* e *software*.
- Alto custo do backup - múltiplos pontos de armazenamento.
- **O problema**:

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Equipamentos ociosos.
- Custo: manutenção de *hardware* e *software*.
- Alto custo do backup - múltiplos pontos de armazenamento.
- **O problema**:
  - Desperdício.
- **A solução**:

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Equipamentos ociosos.
- Custo: manutenção de *hardware* e *software*.
- Alto custo do backup - múltiplos pontos de armazenamento.
- **O problema**:
  - Desperdício.
- **A solução**:
  - Aluguel da infraestrutura.

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Investimento em mais equipamento ?

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Investimento em mais equipamento ?
- Computação na nuvem - aluguel / empréstimo

Computação na nuvem

Armazenamento de dados unificado e Compartilhamento de processamento

- Investimento em mais equipamento ?
- Computação na nuvem - aluguel / empréstimo
  - processamento
  - armazenamento
  - 100% conectado

Computação na nuvem

Investimento

- O provedor
  - Manutenção: hardware e software.
  - Atualizações.
  - Mão de obra especializada.
  - Estar conectado
- O cliente:
  - Sistema de assinatura.
  - Estar conectado

Computação na nuvem

- O que computação em nuvem ?

Computação na nuvem

- O que computação em nuvem ?

Definição

Computação em Nuvem é um modelo do tipo 'pague pelo uso' para possibilitar acesso de rede disponível, conveniente e sobre demanda a um pool compartilhado de recursos computacionais configuráveis (e.g., servidores, armazenamento, redes, aplicações, serviços) que podem ser rapidamente aprovisionados e liberados com o mínimo esforço gerencial ou de interação de provedor de serviços.

National Institute of Standarts and Technology

Computação na nuvem

Principais tipos de serviço

- IaaS: *Infrastructure as a Service*.
- SaaS: *Software as a Service*.
- PaaS: *Platform as a Service*.

Computação na nuvem

IaaS - *Infrastructure as a Service*

- Fornecimento de um ambiente computacional completo:
  - Processamento.
  - Armazenamento de dados.
  - Recursos de rede (servidores).
- Flexibilidade conforme a necessidade do cliente

Computação na nuvem

IaaS - *Infrastructure as a Service*

- Fornecimento de um ambiente computacional completo:
  - Processamento.
  - Armazenamento de dados.
  - Recursos de rede (servidores).
- Flexibilidade conforme a necessidade do cliente

Computação na nuvem

SaaS - *Software as a Service*

- Grande economia tanto em hardware como em software.
- Preço relativamente baixo pela utilização.
- Provedor investe mais no desenvolvimento do produto.
- Modelo de negócio - aluguel $\times$ licença.
- Solução para pirataria de *software*.

Computação na nuvem

PaaS - *Platform as a Service*

- Ambiente de desenvolvimento completo.
  - Ide's, compiladores, banco de dados, etc...
- Flexibilidade de utilização das ferramentas.
- Sistemas operacionais facilmente trocados, atualizados.
- Livre escolha de tudo nas máquinas fornecidas.
- Local de trabalho concentrado.
- Investimento concentrado.

Computação na nuvem

- Tipos de nuvens:
  - Privadas.
  - Públicas.
  - Comunitárias.
  - Híbridas.

Computação na nuvem

- Características:
  - Auto-atendimento sob demanda.
  - Amplo acesso a rede.
  - Pool de recursos.
  - Elasticidade rápida.
  - Serviços Mensuráveis.

Computação na nuvem

Fog computing

- Infraestrutura na nuvem e na própria rede:
  - Melhorar a eficiência e reduzir a quantidade de dados que precisam ser transmitidos para que seja feito o processamento, análise e armazenamento.
  - Aproximar o consumidor de informações com o provedor de dados.
  - Prover segurança e conformidade para a transmissão de dados.

Computação na nuvem

Fog computing

- Infraestrutura na nuvem e na própria rede:
  - Melhorar a eficiência e reduzir a quantidade de dados que precisam ser transmitidos para que seja feito o processamento, análise e armazenamento.
  - Aproximar o consumidor de informações com o provedor de dados.
  - Prover segurança e conformidade para a transmissão de dados.

Computação na nuvem

Fog computing

- Características:
  - Conhecimento de localização e baixa latência de comunicação.
  - Dispositivos localizados em ambiente altamente distribuído.
  - Grande volume de dispositivos, como uma consequência da distribuição geográfica.
  - Suporte a mobilidade, como por exemplo, carros conectados a rede, *smartphones*, *smartwatches* e etc.
  - Processamento em tempo real.
  - Acesso predominantemente via *Wireless*.
  - Heterogeneidade. Dispositivos *fog* tendem a ser os mais variados, com diferentes tipos de dados gerados, protocolos de comunicação, entre outros.
  - Interoperabilidade e virtualização. Algumas situações, como por exemplo transmissão de vídeo em tempo real, requerem a cooperação de diferentes provedores. Nesse sentido, *fog computing* precisa ser capaz de interoperar e os serviços precisam ser virtualizados entre os diferentes domínios.

Computação na nuvem

Edge Computing

- Metodologia de processamento:
  - Processamento na nuvem $\times$ processamento na ponta (celular e dispositivos embarcados).

Hardware × Software

Computação de alto desempenho

Hardware × Software

- Computadores pessoais com capacidade de processamento paralelo
- Multi-cores, many-cores e distribuídos
- Códigos paralelos e/ou distribuídos

Computação de alto desempenho

Hardware × Software

- Computadores pessoais com capacidade de processamento paralelo
- Multi-cores, many-cores e distribuídos
- Códigos paralelos e/ou distribuídos

Arquiteturas híbridas !

Computação de alto desempenho

Hardware × Software

- Qual a diferença entre paralelo e distribuídos ?
- Meu código é paralelo e/ou distribuído ? É eficiente ?

Computação de alto desempenho

Hardware × Software

- x86:
  - OpenMP (*Open Multi-Processing*) e *pthread*
- GPU:
  - CUDA - (*Compute Unified Device Architecture*)
  - OpenCL (*Open Computing Language*)
- *Clusters*:
  - MPI - (*Message Passing Interface*)

OpenMP - Open *Multi-Processing*

OpenMP - Open *Multi-Processing*

- Necessidade de um padrão para processamento paralelo de memória compartilhada
- Histórico:
  - 1997 - versão 1.0 para FORTRAN
  - 1998 - versão 1.0 para C/C++
  - 2000 - versão 2.0
  - 2008 - versão 3.0

OpenMP - Open *Multi-Processing*

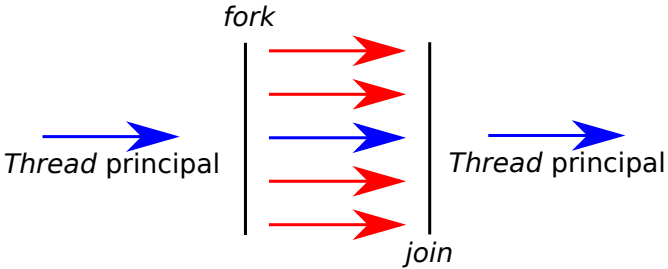
- Incorporado no compilador - uso através de diretivas
- Sentinela
  - Combinação de caracteres que definem a diretiva
  - Exemplo em C/C++: `#pragma omp`
- Ignora a diretiva caso não reconheça
- Uso de blocos `Parallel OMP`

OpenMP - Open *Multi-Processing*

- Permite um alto nível de abstração
- Modelo de memória compartilhada
- *Thread-safe* de dados é simplificado
- Inicia a execução do programa com uma *thread* principal
- A *thread* principal cria *threads* trabalhadoras
  - Garga de trabalho e as *threads* trabalhadoras
- *Threads* trabalhadoras terminam a execução fim da região paralela
- Usa sincronização para evitar a concorrência entre as *threads*

OpenMP - Open *Multi-Processing*

- Modelo de execução tipo *fork-join*
- **Thread-principal** → região paralela → **Thread-principal**
  - Apenas a thread principal continua executando após a região paralela
  - Sincronização é implícita



OpenMP - Open *Multi-Processing*

- Trabalha com diretivas de compilação:
  - Comandos que são ativados apenas pelos compiladores
  - Suporte as diretivas pelo compilador
    - FORTRAN, C e C++
  - Exemplo: `gcc -fopenmp arquivo.c -o hello`

CUDA - *Compute Unified Device Architecture*

CUDA - *Compute Unified Device Architecture*

- História: GPU (*Graphics Processing Unit*)
  - Antes de 2008: Pipeline gráfico (fixo e programável)
  - Depois de 2008: Arquitetura Unificada
  - 2009: OpenCL - *Open Computer Language*

CUDA - *Compute Unified Device Architecture*

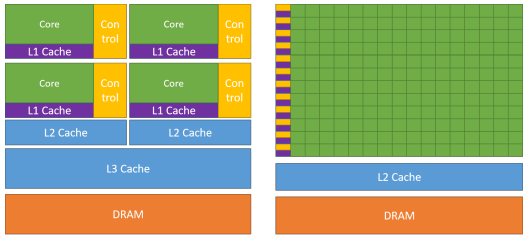
- As GPUs são usadas na industria:
  - Óleo e gás: processamento sísmico, simulação de reservatório, etc...
  - Pesquisa: astrofísica, dinâmica dos fluídos, previsão de tempo, etc...
  - Defesa e governo: processamento de imagem satélite, processamento de imagens e videos, etc...
  - Simulação bio-química, seqüência de DNA, etc...
  - Finanças: Método de Monte Carlo, análise de risco financeiro, etc...
  - Produção: Mecânica Estrutural, etc...
- Soluções de arquiteturas dedicadas e desenvolvimento de tecnologia.

CUDA - *Compute Unified Device Architecture*

- Capacidade de processamento de ponto flutuante.
  - 16 bits, 32 bits e 64 bits
- Uso da GPU para processamento não gráfico.
- GPGPU - *General Purpose computation on* GPU: Uso da GPU para programação de propósito geral usando alguma API gráfica.
- GPU *Computing*:Uso da GPU para computação paralela através de linguagem de programação e API, sem uso de APIs gráficas.
- CUDA: Modelo de programação paralela e plataforma de software para GPU.

CUDA - *Compute Unified Device Architecture*

- Unidade de controle de fluxo de instruções.
- Tamanho das memórias (Cache e MP).
- ALUs e cores.
- Gerência e criação de *threads*.



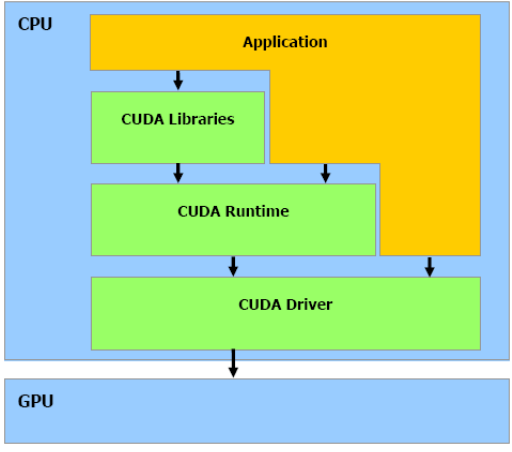
Fonte: (https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html) - acessado em 15 de set de 2020

CUDA - *Compute Unified Device Architecture*

- Paralelismo extensivo de dados: milhares de computações feitas sobre os dados.
- Grupos de tarefas simples: grupo de *threads* podem executar diferentes programas.
- Aritmética intensiva de ponto flutuante.
- Latência e tolerância: quantidade de tarefas executas por instantes de tempo.
- Fluxo de dados: alta largura de banda para transferência dos dados com pouco reuso.
- Sincronismo entre *threads* de forma simples (barreira). *Threads* de um mesmo bloco.
- Custo zero para escalonamento das *threads*.

CUDA - *Compute Unified Device Architecture*

- Biblioteca (APIs).
  - Aplicação.
  - *runtime*.
  - *driver*.
- *toolkit*.
- *driver* da placa gráfica.



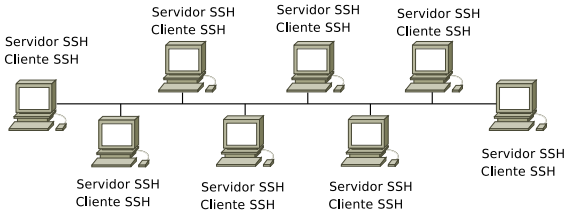
CUDA - *Compute Unified Device Architecture*

- Conceitos:
  - GPU: Dispositivo de computação com capacidade de executar *threads* em paralelo.
  - CPU: *Host* que trabalha em conjunto com a GPU.
  - *Kernel*: instância do programa.
  - *Thread*: instância de um *Kernel*.
  - Memória de vídeo / *device*.
  - Memória principal / *host*.

MPI - (*Message Passing Interface*)

MPI - (*Message Passing Interface*)

- Computadores distribuídos:
  - Processador
  - Memória
  - Todos os subsistemas computacionais
  - Comunicação por troca de mensagem



MPI - (*Message Passing Interface*)

- Meu próprio *cluster* computacional!
  - Quando se utilizam dois ou mais computadores em conjunto para resolver um problema, você tem um *cluster*, que do inglês significa agrupamento

MPI - (*Message Passing Interface*)

- Alta Disponibilidade (HA - *High Availability*)
  - Os *clusters* HA têm a finalidade de manter um determinado serviço de forma segura o maior tempo possível.
- **Alto Desempenho (HPC - *High Performance Computing*)**
  - Configuração voltada para prover o maior poder computacional possível.

MPI - (*Message Passing Interface*)

- Vantagens dos *clusters* computacionais
  - Alto Desempenho
  - Escalabilidade
  - Tolerância a Falhas
  - Baixo custo
  - Independência dos fornecedores

MPI - (*Message Passing Interface*)

- Classificação segundo prioridades
  - dedicados: utilizam seus nós exclusivamente para computação paralela
  - não-dedicados: as aplicações são executadas baseadas na ociosidade das estações de trabalho

MPI - (*Message Passing Interface*)

- *Cluster Beowulf*
  - 16 computadores pessoais - microprcessador 486
  - Linux
  - Rede padrão Ethernet (10Mbps)
  - Poder computacional de 70 megaFLOPS
  - Custo: aproximadamente US$ $50,000.00$

MPI - (*Message Passing Interface*)

- *Cluster Beowulf*
  - 16 computadores pessoais - microprcessador 486
  - Linux
  - Rede padrão Ethernet (10Mbps)
  - Poder computacional de 70 megaFLOPS
  - Custo: aproximadamente US$ $50,000.00$

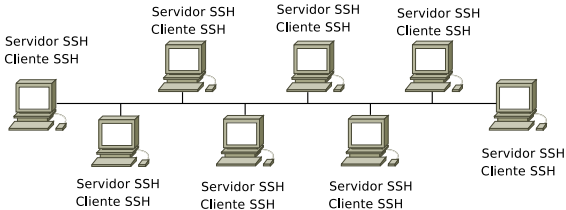
“Faça você mesmo o seu próprio supercomputador”

MPI - (*Message Passing Interface*)

- *Cluster* - classe de máquina
  - Classe 1: gerencia as tarefas
  - Classe 2: executa as tarefas

MPI - (*Message Passing Interface*)

- *Cluster* - classe de máquina
  - Classe 1: gerencia as tarefas
  - Classe 2: executa as tarefas



MPI - (*Message Passing Interface*)

- *Cluster* componentes:
  - Nó ou node
  - SO
  - Rede Local
  - Protocolos
  - *cluster middleware* - MPI
  - Sistemas de Arquivos Paralelos

MPI - (*Message Passing Interface*)

- As máquinas do *Cluster* podem ser:
  - Homogêneas: todas as máquinas são iguais
  - Heterogêneas: máquinas diferentes

FPGA

FPGA

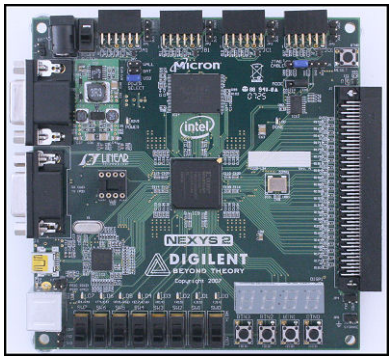
Introdução

- FPGA - ***F**ield **P**rogrammable **G**ate **A**rrays*
  - Programação de circuitos - HDL
  - Primeiro, vieram PROMs (*Programmable Read Only Memories*) e PLDs (*Programmable logic devices*), matrizes de portas (re-)configuráveis.
  - Algumas patentes de coisas parecidas com FPGAs surgiram no final dos anos 80 e início dos anos 90.
  - 1985 - comercializado o XC2064, primeira FPGA comercial.
    - O XC2064 tinha 64 blocos lógicos configuráveis e interconexões configuráveis entre os blocos lógicos
    - O XC2064 só tinha blocos lógicos configuráveis (CLBs), cada um com duas LUTs de 3 entradas

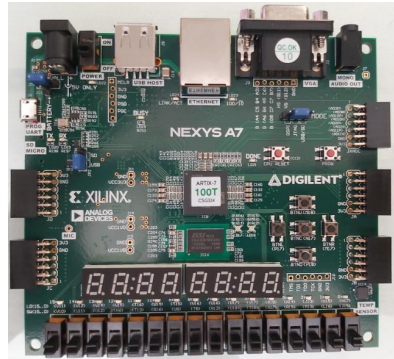
FPGA

Introdução

- Nexys 2



- Nexys A7



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
UFRRJ

Medidas de desempenho

Medidas de desempenho

- tempo de execução da aplicação (tempo de parede)
  - Não precisa utilizar qualquer biblioteca
  - *Speedup* = $\frac{T_s}{T_p}$
  - Forma mais simples
- eficiência: $\frac{T_s}{P \times T_p}$
- *Overhead*: inimigo da performance
- ciclos de clock
  - Requer uso de alguma ferramenta ou biblioteca externa
  - Forma mais simples
- quantidade de operações de ponto flutuante/instruções por unidade de tempo
  - Uso de alguma ferramenta ou biblioteca externa
  - Processador permita a coleta do evento

Medidas de desempenho

- Definir e documentar a metodologia de execução
- Salvar log com os resultados - CSV
- Trabalhar com a máquina dedicada
- Fixar o core de execução da *thread*
- Fixar a frequência do processador

Medidas de desempenho

Exemplo: problema dos n-corpos

Saída: Posição das n particulas no tempo $t+1$

1 **for** $i \leftarrow 1$ ***TO*** n **do**

2 $\quad F_i^{t+1} \leftarrow -G \sum_{j=1}^{n} \frac{m_i \times m_j}{d_{i,j}^2} \times \frac{p_i^t - p_j^t}{d_{i,j}} \ \forall i \neq j;$

3 $\quad v_i^{t+1} \leftarrow v_i^t + \Delta t \times F_i^{t+1};$

4 $\quad p_i^{t+1} \leftarrow p_i^t + \Delta t \times v_i^{t+1};$

5 **end**

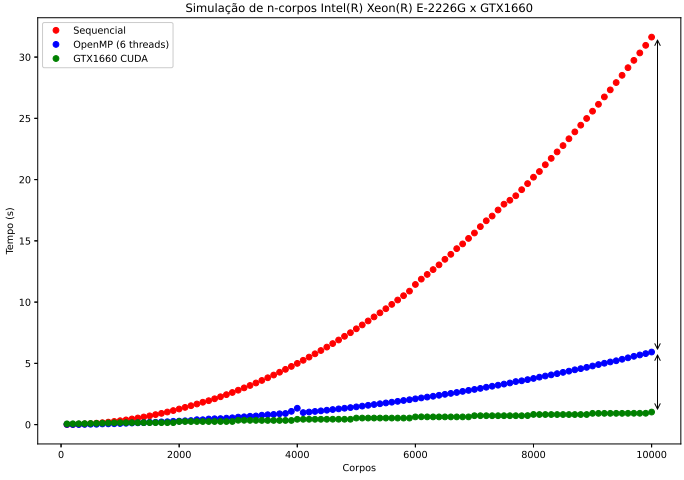
Algoritmo 1: Algoritmo dos n-corpos - partícula a partícula.

- Condição inicial - posição aleatória e velocidade zero.
- Constante gravitacional $G = 1$.
- $\Delta t = 1.5$.
- 100 passos de tempo. Média de dez execuções.
- Partículas entre 10^2 até 10^4, incluindo 10^2 partículas por simulação.

- CPU: Intel(R) Xeon(R) E-2226G CPU @ 3.40GHz/6 cores físicos/12MB cache L3
- GPU:GTX 1660 - 1800 MHz/1408 cuda-cores em 22 SMP/6GB

Medidas de desempenho

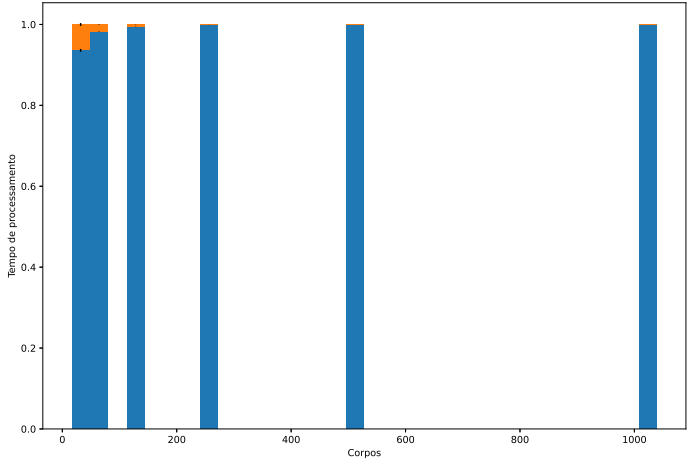
Exemplo: problema dos n-corpos



- OpenMP:
  - $S = \frac{Ts}{Tp} = \frac{31.63}{5.92} = 5.34$
  - $E = \frac{Ts}{P \times Tp} = \frac{31.63}{6 \times 5.92} = 0.98$
- CUDA:
  - $S = \frac{Ts}{Tp} = \frac{31.63}{1.03} = 30.84$
  - $E = \frac{Ts}{P \times Tp} = \frac{31.63}{1408 \times 1.03} = 0.0007$

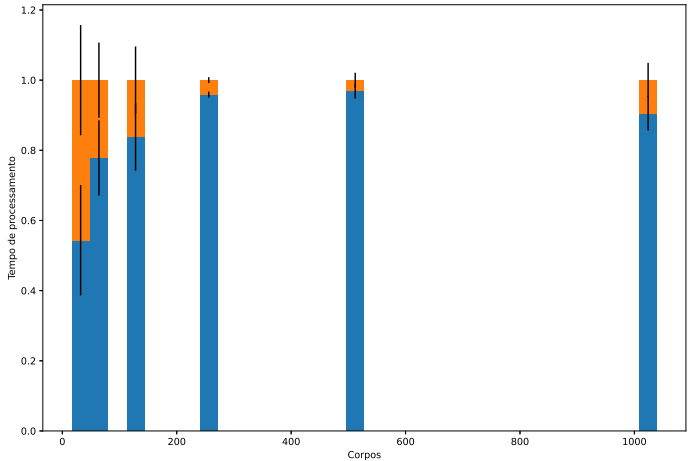
Medidas de desempenho

Exemplo: problema dos n-corpos



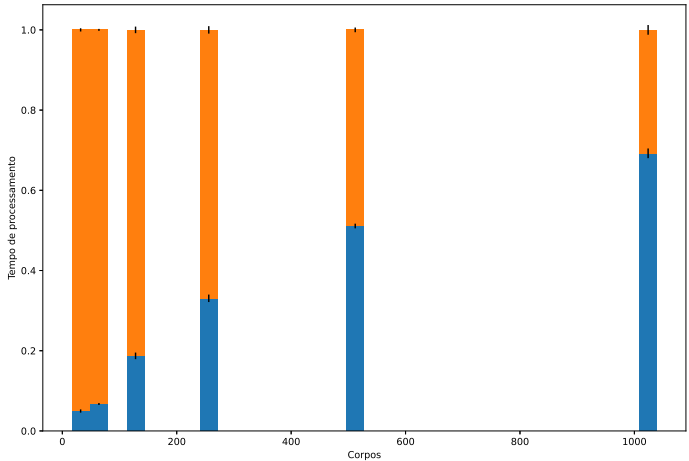
Medidas de desempenho

Exemplo: problema dos n-corpos



Medidas de desempenho

Exemplo: problema dos n-corpos



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
UFRRJ