

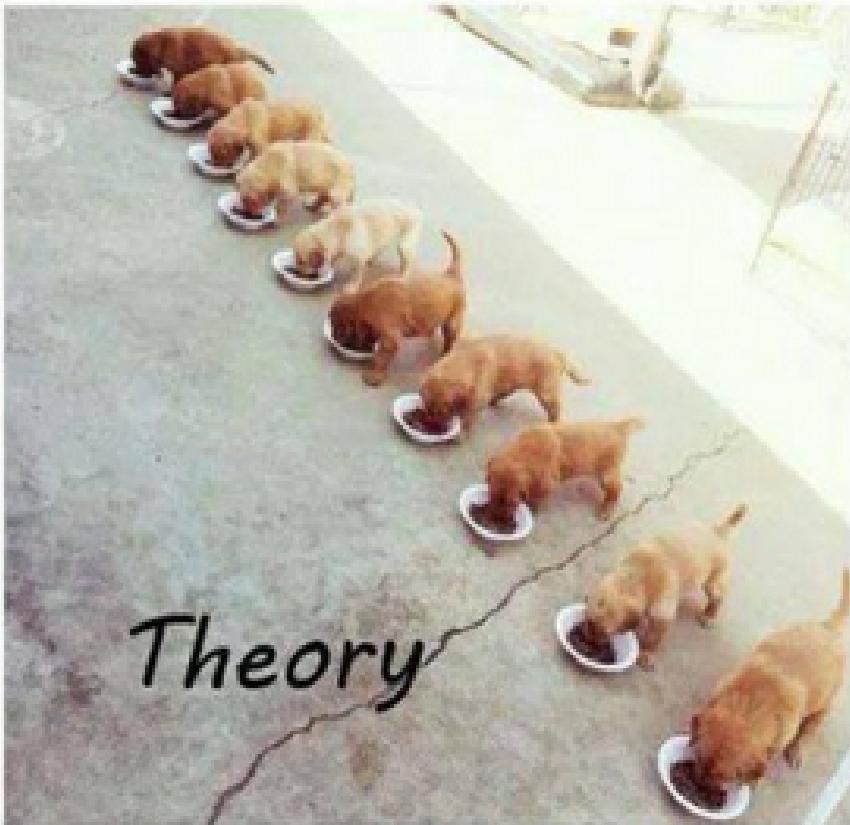
Programação Paralela – OPRP001

Aplicabilidade do Paralelismo

Desenvolvido por: Prof. Guilherme Koslovski e Prof. Maurício A. Pillon

Então...

Multithreaded programming



Comentários iniciais

- 👉 Avaliação
 - **Nota Final = A1 * 0,30 + A2 * 0,35 + A3 * 0,35**
- 👉 Plano de ensino:
 - Aulas presenciais c/ lista de presença;
 - Correio Eletrônico
 - Plataforma Moodle (<http://moodle.joinville.udesc.br/>)
 - Escritório do professor
- 👉 Exame final
- 👉 Como estudar?

Bibliografia

- **DE ROSE, César A. F.; NAVAUX, Philippe O. A. Arquiteturas Paralelas. Porto Alegre: Sagra-Luzzato, 2003**
- FOSTER, Ian. Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Editora Addison-Wesley, 1995
- **COMISSÃO REGIONAL DE ALTO DESEMPENHO - RS. Caderno dos Cursos Permanentes. Porto Alegre: SBC, 2006.**
- STALLINGS, William. Operating Systems - Internals and Design Principles. Prentice-Hall. 1997. 3a. Edição
-

Agenda

- ↳ **Motivação**

- ↳ **Definições**

- ↳ Classificação de arquiteturas paralelas

- ↳ Exemplos de arquiteturas paralelas

- ↳ Estudo de caso

Execução de um programa

- ↳ Sequencial
- ↳ Diversos processadores? GPU?
- ↳ Onde está a informação (dados)?
- ↳ Fator limitante: hardware ou software?

Motivação e definições

- Um computador paralelo é um conjunto de processadores que estão habilitados para resolver um problema [Ian Foster].
- Programação paralela é a divisão de uma determinada aplicação em partes, de maneira que essas partes possam ser executadas simultaneamente, por vários elementos de processamento [ERAD 2012].

Motivação e definições

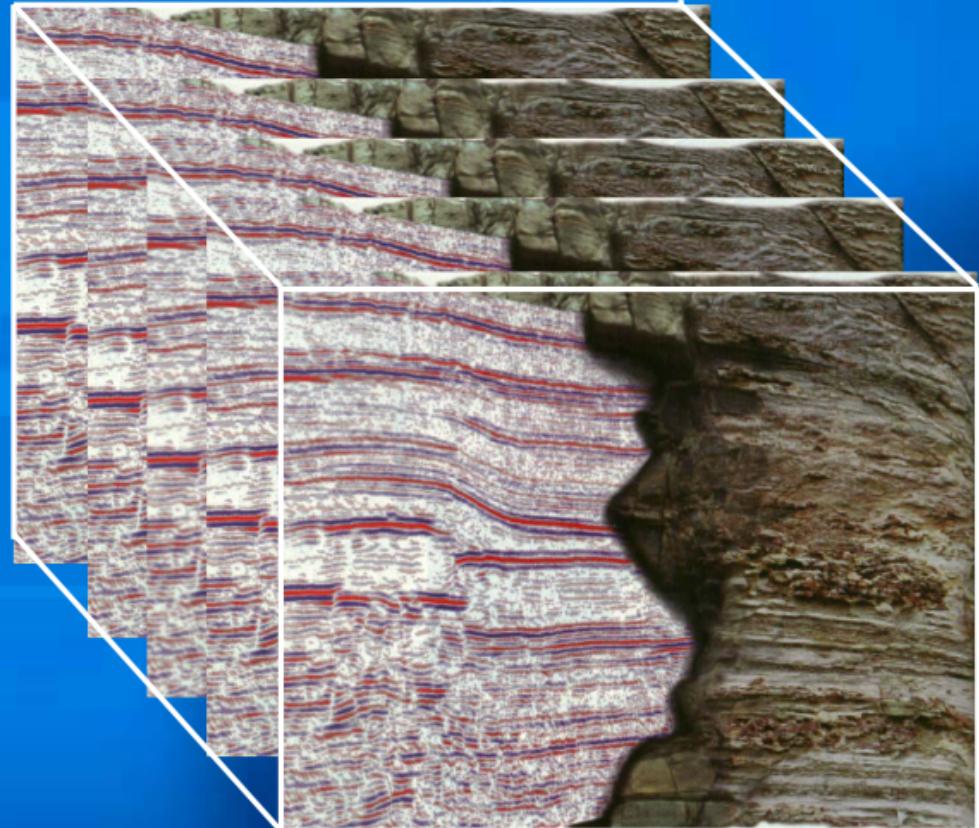
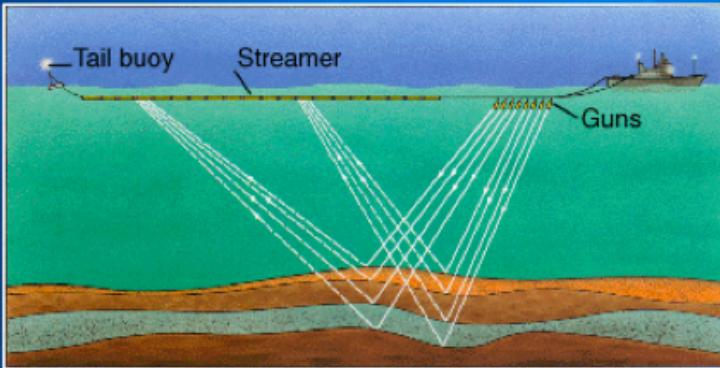
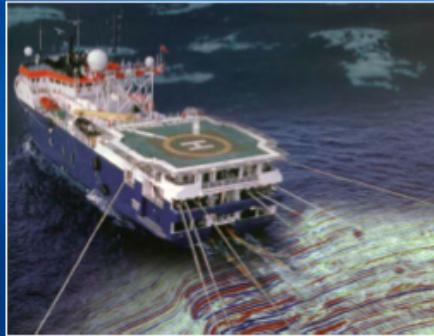
- ☞ Aplicações complexas
- ☞ Tempo de execução sequencial inviável
- ☞ Exemplos
 - ☛ Simulações de meteorologia
 - ☛ Simulações sobre pontos de perfuração de poços de petróleo
 - ☛ Simulações aerodinâmicas
 - ☛ Análise de algoritmos de criptografia
 - ☛ Processamento do Imposto de Renda
 - ☛ Desenvolvimento de filmes, edição de vídeos
 - ☛ Simulações sobre energia nuclear
 - ☛ Simulações moleculares
 - ☛ Avaliação de protótipos de novas tecnologias
 - ☛ ...

Motivação e definições

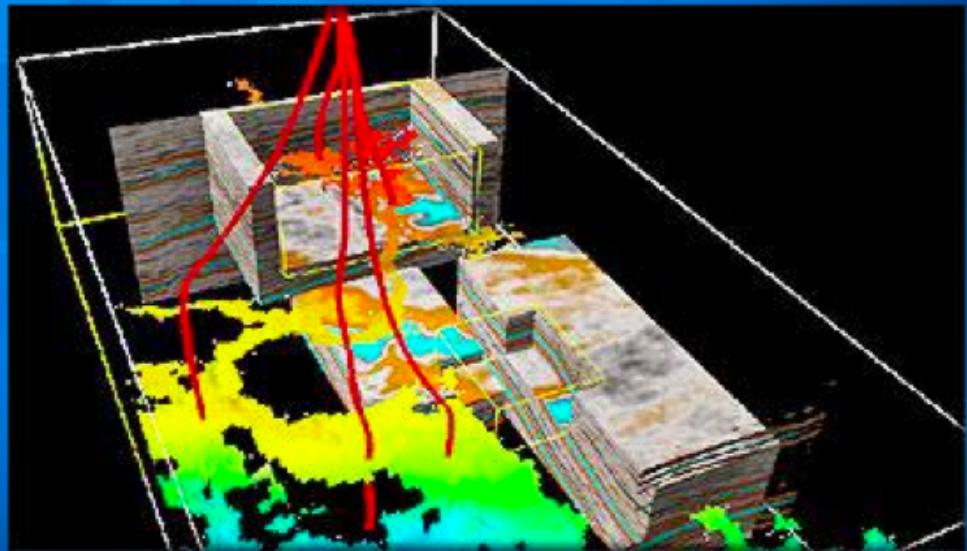
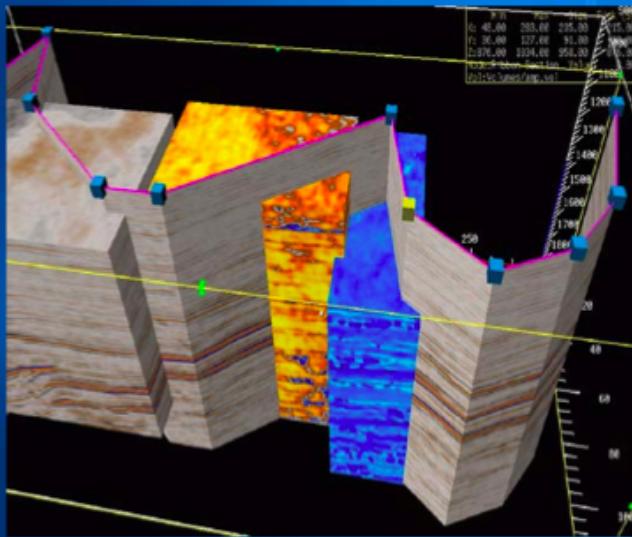
- ↳ Exemplos de aplicações

- ↳ Sistemas bancários
 - ↳ Vídeo conferência
 - ↳ Motores de busca (google)
 - ↳ Realidade virtual
 - ↳ Circuitos eletrônicos
 - ↳ Reações químicas
 - ↳ ...

Exemplo Petrobras



Exemplo Petrobras



Exemplo Petrobras

#	Nome	Marca	País	Área	Power (KW)
1	Sequoia	IBM	USA	Research	7.890
2	K Computer	Fujitsu	Japan	Research	12.660
3	Mira	IBM	USA	Research	3.945
4	SuperMUC	IBM	Germany	Academic	3.423
5	Tianhe-1A	NUDT	China	Research	4.040
6	Jaguar	Cray	USA	Research	5.142
7	Fermi	IBM	Italy	Academic	822
8	JuQUEEN	IBM	Germany	Research	658
9	Curie	Bull SA	France	Research	2.251
10	Nebulae	Dawning	China	Research	2.580
68	Grifo04	Itautec	PETROBRAS	Industry	366
79	Tupã	Cray	INPE	Research	
457	Galileu	Sun/Oracle	UFRJ	Academic	430

Exemplo Santos Dumont



<http://sdumont.lncc.br/machine.php?pg=machine#>

Tupã: INPE



- Tupã na mídia:
 - [Reportagem Jornal Nacional sobre Tupã \(28/12/2010\)](#)
 - [Reportagem Bom Dia Brasil sobre Tupã \(30/01/2017\)](#)
- Introdução ao uso do Tupã
- O sistema de supercomputação Cray XE6 do INPE é composto por:
 - 13 nós de acesso interativo;
 - 128 GB (8GB por core); 4 CPU quad-core AMD Opteron; Ethernet 1 e 10Gigabit
 - 20 nós de processamento auxiliar;
 - Possuem 2.56 TB de memória total agregada;
 - 1304 nós computacionais Cray XE6; e
 - 32 nós de serviço.

Tupã: INPE



MODELOS DE PROGRAMAÇÃO PARALELA

Vários tipos de paralelismo podem ser explorados no Tupã utilizando diferentes modelos e métodos de programação, conforme listados na tabela abaixo:

Nível de Hardware	Modelo	Descrição
Shared-memory node	Auto	Habilita parallelização automática em programas de memória compartilhada. Loops básicos podem ser automaticamente paralelizados, mas para melhoria adicional de desempenho, tente utilizar diretivas OpenMP . Esta opção está disponível nos ambientes PGI e Pathscale .
Shared-memory node	OpenMP	Esta é uma forma de programação paralela explícita na qual o programador insere diretivas no programa para disparar múltiplos threads de memória compartilhada, tipicamente a nível de loop. É um método comum de parallelização, portável e relativamente fácil de implementar. Por outro lado, ele requer memória compartilhada cujos limites escalam ao número de processadores em apenas um nó. OpenMP pode ser utilizado em conjunto com a autoparalelização e está disponível com os ambientes PGI , Pathscale e GNU .
Distributed memory	MPI	Este é o método mais comum e portável para parallelização de códigos para sistemas de memória distribuída escalável. MPI é uma biblioteca de subrotinas para troca de mensagens, operações coletivas e outras formas de comunicação inter-processadores. O programador é responsável por implementar a distribuição de dados, sincronização e remontagem dos resultados utilizando chamadas explícitas MPI . Utilizando MPI , o programador pode amplamente ignorar a organização física dos processadores nos nós e simplesmente tratar o sistema como uma coleção de processadores independentes. MPI está disponível com os ambientes PGI , Pathscale e GNU .
Distributed memory	SHMEM	Esta biblioteca suporta comunicações <i>one-sided</i> entre as tarefas.

Tupã: INPE



AMBIENTE DE PROGRAMAÇÃO

Estão disponíveis no Tupã as seguintes ferramentas de programação. Certifique-se de trocar para o *module* apropriado quando alternar entre os compiladores.

Item	PGI	PathScale	GNU Compilers
Fortran 77/90/95 (Compute Node)	<i>ftn</i>	<i>ftn</i>	<i>ftn</i>
C compiler (Compute Node)	<i>cc</i>	<i>cc</i>	<i>cc</i>
C++ compiler (Compute Node)	<i>CC</i>	<i>CC</i>	<i>CC</i>
Fortran 77/90/95 (Login Node Only)	<i>pgf90</i>	<i>pathf90</i>	<i>gfortran</i>
C compiler (Login Node Only)	<i>pgcc</i>	<i>pathcc</i>	<i>gcc</i>
C++ compiler (Login Node Only)	<i>pgCC</i>	<i>pathCC</i>	<i>g++</i>
Debuggers	<i>totalview</i>	<i>totalview</i>	<i>totalview</i>
Performance Analysis	<i>CrayPat</i>	<i>CrayPat</i>	<i>CrayPat</i>
Default module (*)	<i>PrgEnv-pgi</i> (default)	<i>PrgEnv-pathscale</i>	<i>PrgEnv-gnu</i>
Batch queueing system		<i>PBS Professional 10.2</i>	

NOTA: O ambiente de programação PGI é carregado por padrão para todas as contas.

Motivação: concursos

1 - Q25834 (*Prova: CESGRANRIO - 2012 - Chesf - Profissional de Nível Superior - Analista de Sistemas / Arquitetura de Computadores / Sistemas Distribuídos;*)

Um sistema, rodando em um computador com processamento paralelo do tipo single-instruction multiple data (SIMD), não pode ser considerado como um sistema distribuído.

PORQUE

Sistemas distribuídos consistem em uma coleção interconectada de processadores e/ou computadores autônomos, conhecidos como nós, com o propósito de executar uma ou mais aplicações.

Analizando-se as afirmações acima, conclui-se que

- a)** as duas afirmações são verdadeiras, e a segunda justifica a primeira.
- b)** as duas afirmações são verdadeiras, e a segunda não justifica a primeira.
- c)** a primeira afirmação é verdadeira, e a segunda é falsa.
- d)** a primeira afirmação é falsa, e a segunda é verdadeira.
- e)** as duas afirmações são falsas.

Motivação: concursos

3 - Q202781 (*Prova: IADES - 2011 - PG-DF - Analista Jurídico - Analista de Sistemas / Arquitetura de Computadores / Sistemas Distribuídos;*)

Segundo Andrew Tanembaum (2007) "Sistema Distribuído é uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente". Assinale a alternativa correta a respeito de um sistema de informação distribuído.

- a)** A distribuição de tarefas se dá a partir de requisições do usuário, que indica o endereço do servidor onde deseja executar tal tarefa.
- b)** Em uma rede de computadores há servidores dedicados a atender pedidos dos clientes e estes, por sua vez, têm função exclusiva de requisitantes.
- c)** Todos os computadores de uma rede executam tarefas de cliente e servidor, quando se deseja integrá-los em uma arquitetura de sistemas distribuídos.
- d)** A transparência de acesso é uma característica dos sistemas distribuídos que permite que recursos sejam acessados sem que sua localização seja determinada.
- e)** Em um sistema de objetos distribuídos é possível invocar métodos de um objeto, ainda que este não esteja presente no computador do usuário.

5 - Q106388 (*Prova: CESPE - 2011 - FUB - Técnico de Tecnologia da Informação - Específicos / Arquitetura de Computadores / Sistemas Distribuídos;*)

Acerca dos conceitos básicos e modos de utilização da informática, julgue os itens a seguir

A computação em grade (*grid computing*) constitui solução para a interligação de servidores ou estações de trabalho para simular um único supercomputador formado por nodos.

Certo Errado

Motivação: concursos

8 - Q10908 (*Prova: CESGRANRIO - 2008 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura / Arquitetura de Computadores / Conceitos Básicos; Sistemas Distribuídos;*)

Sobre o processamento paralelo e distribuído, assinale a afirmação correta.

- a)** A computação paralela é caracterizada pelo uso de vários processadores para executar uma computação de forma mais rápida, baseando-se no fato de que o processo de resolução de um problema pode ser dividido em tarefas menores, que podem ser realizadas simultaneamente através de algum tipo de coordenação.
- b)** A execução de tarefas em um ambiente de processadores distribuídos com acoplamento fraco prevê que a memória seja compartilhada entre os processos trabalhadores.
- c)** Em programação paralela não é necessário se conhecer a arquitetura de comunicação entre processadores para elaborar os programas.
- d)** Um grid computacional pode ser formado por diversos computadores, heterogêneos, que não podem estar distribuídos via Web por problemas de segurança.
- e)** Um sistema distribuído fortemente acoplado é formado por um ambiente de computadores dotados de memória e sistema operacional próprios, que se comunicam via switch.

Motivação e definições

- ↳ Em resumo
 - ↳ Potência de cálculo com recursos computacionais concentrados
 - ↳ Cada vez mais presente e próximo dos usuários finais
- ↳ Clusters e Grids
- ↳ E Cloud Computing?
 - ↳ Google
 - ↳ Facebook
 - ↳ Amazon EC2
 - ↳ ...

Definições

- 👉 **Sistemas Distribuídos**
 - 👉 Mais geral e universal que a computação paralela (grande área)
- 👉 **Sistemas Paralelos**
 - 👉 Paralelismo é uma forma restrita de computação distribuída
 - 👉 Todo o sistema está dedicado a solução de um único problema
- 👉 **Sistemas Concorrentes**
 - 👉 Ações que podem ser executadas simultaneamente. O programa concorrente especifica dois ou mais programas sequenciais que podem ser executados como processos paralelos
- 👉 Ponto em comum: todos os sistemas requerem a presença de múltiplos processadores