并行程序设计概述

周　斌 @ NVIDIA & USTC 2014年8月

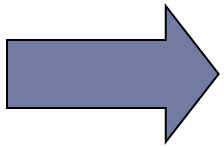
内容

- 为什么需要?
- 怎么做?
- 一些技术和概念

并行处理在生活中常见

- 搬砖模型
- 一堆砖头
- 需要搬走

串行处理



CPU模式？

(数据)并行处理方法

摩尔定律

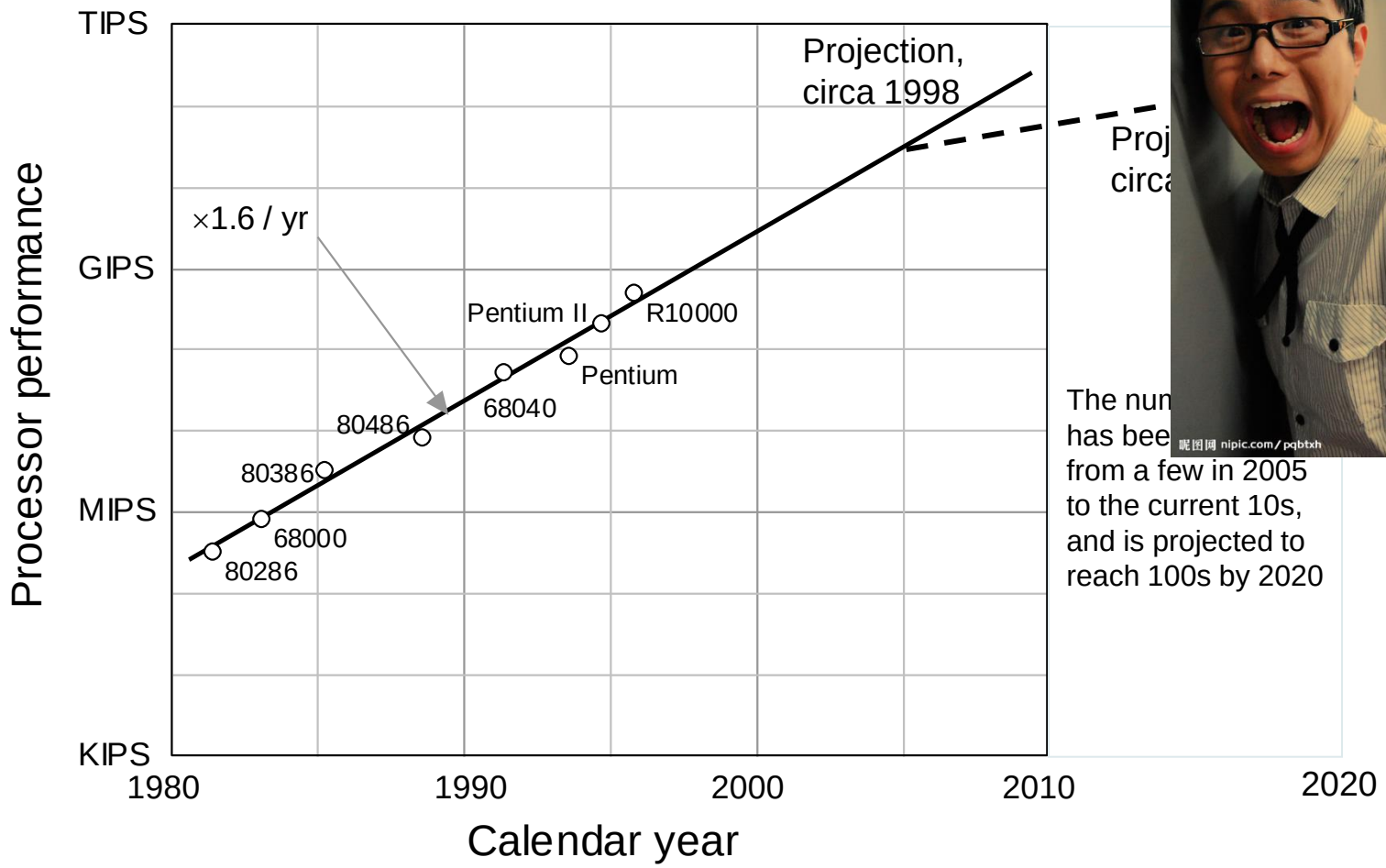
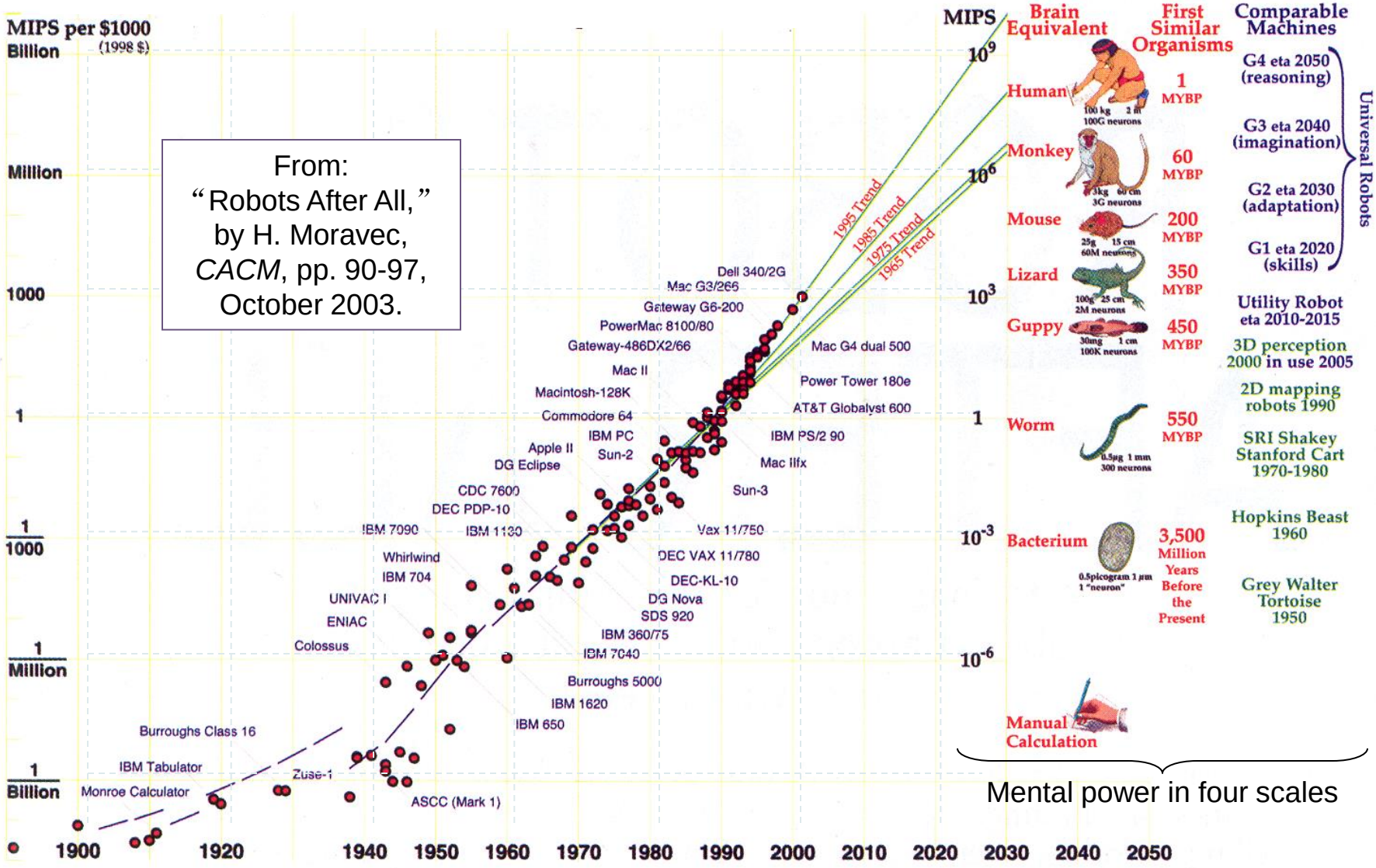


Fig. 1.1　又是摩尔定律 (extrapolated).

计算机性价比变化 Performance/Cost



半导体产业技术路线图

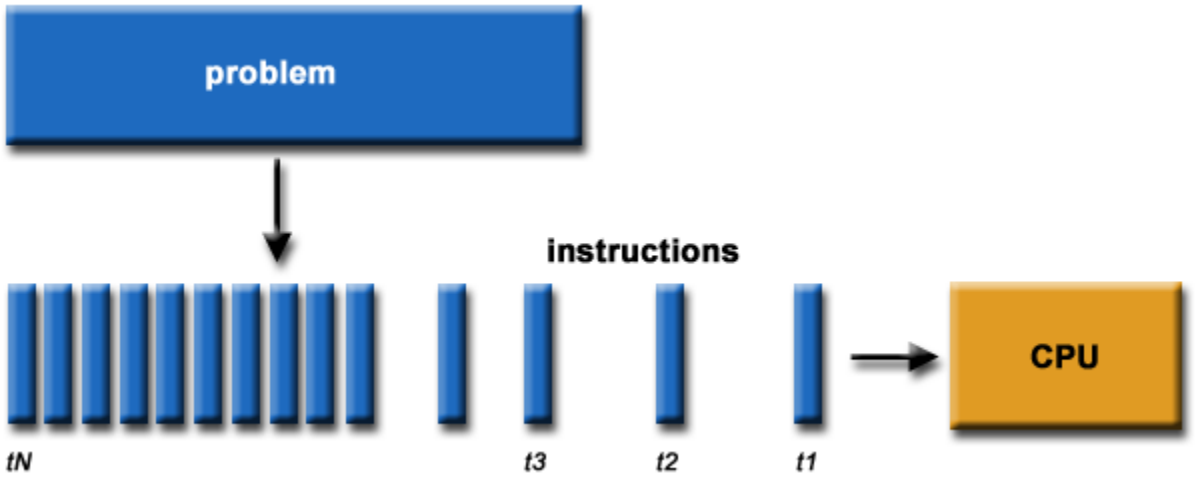
年份 →	2001	2004	2007	2010	2013	2016	2015	2020	2025
制程 (nm)	140	90	65	45	32	22	19	12	8
频率. (GHz)	2	~~4~~	~~7~~	3.6 ~~12~~	4.1 ~~20~~	4.6 ~~30~~	4.4	5.3	6.5
Wiring levels	7	8	9	10	10	10			
电压 (V)	1.1	1.0	0.8	0.7	0.6	0.5			0.6
功耗 (W)	130	160	190	220	250	290			

From the 2001 edition of the roadmap [Alla02]

From the 2011 edition (Executive Summary)

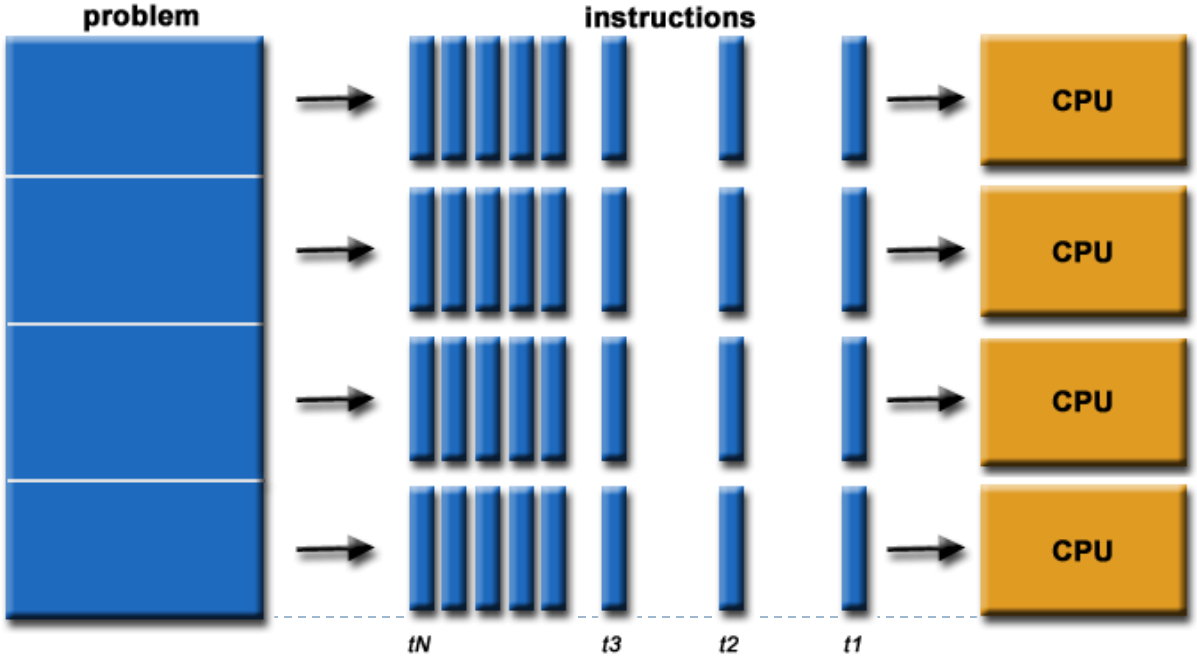
串行计算模式

- 常规软件是串行的
  - 设计运行于一个中央处理器上 (CPU);
  - 通过离散的指令序列完成一个问题的解决
  - 一条一条指令的执行
  - 同时只有一条指令在执行



并行计算模式

- 一句话：并行计算是同时应用多个计算资源解决一个计算问题
  - 涉及多个计算资源或处理器
  - 问题被分解为多个离散的部分，可以同时处理（并行）
  - 每个部分可以由一系列指令完成
- 每个部分的指令在不同的处理器上执行



并行计算可以干什么?

- 太多了.....

- x
- xx
- xxx
- xxxx

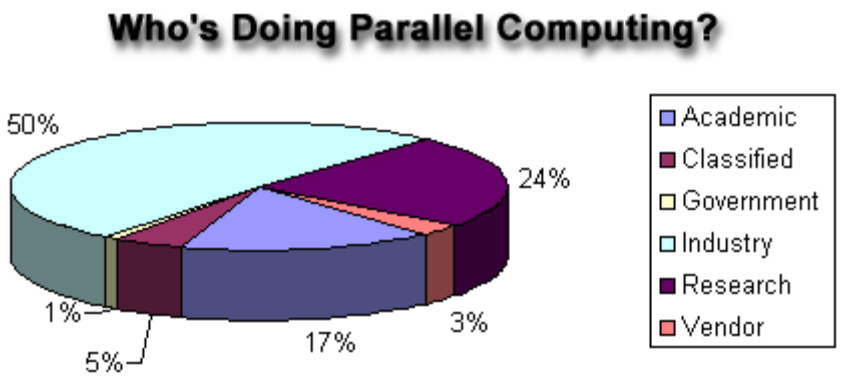
常规理解

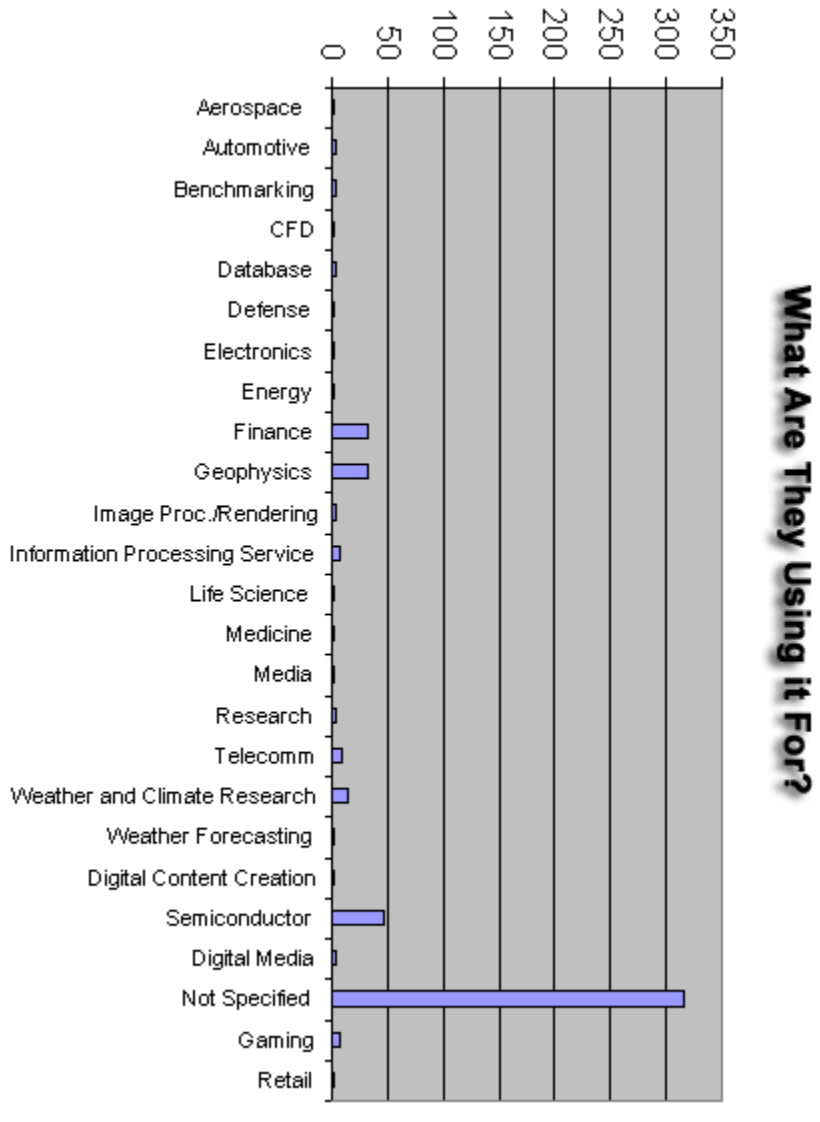
- 高端问题
  - weather and climate
  - chemical and nuclear reactions
  - biological, human genome
  - geological, seismic activity
  - mechanical devices - from prosthetics to spacecraft
  - electronic circuits
  - manufacturing processes

其实已经很深入生活

- 不再是高大上:
  - 数据挖掘
  - 搜索引擎、购物网站
  - 药物
  - 图形图像，游戏
  - 视频，多媒体，虚拟现实
  - 手机
- Ultimately, parallel computing is an attempt to maximize the infinite but seemingly scarce commodity called time.

并行计算分布


Who's Doing Parallel Computing?
50%
24%
1%
5%
17%
3%
Academic
Classified
Government
Industry
Research
Vendor


What Are They Using it For?
0
50
100
150
200
250
300
350
Aerospace
Automotive
Benchmarking
CFD
Database
Defense
Electronics
Energy
Finance
Geophysics
Image Proc./Rendering
Information Processing Service
Life Science
Medicine
Media
Research
Telecomm
Weather and Climate Research
Weather Forecasting
Digital Content Creation
Semiconductor
Digital Media
Not Specified
Gaming
Retail
Data obtained from top500.org, June 2006

概念和名词

简要介绍

Flynn 矩阵

- The matrix below defines the 4 possible classifications according to Flynn

S I S D **Single Instruction, Single Data**	**S I M D** **Single Instruction, Multiple Data**
M I S D **Multiple Instruction, Single Data**	**M I M D** **Multiple Instruction, Multiple Data**

常见名词

- Task（任务）
- Parallel Task （并行任务）
- Serial Execution （串行执行）
- Parallel Execution （并行执行）
- Shared Memory （共享存储）
- Distributed Memory （分布式存储）
- Communications （通信）
- Synchronization （同步）
- Granularity （粒度）
- Observed Speedup （加速比）
- Parallel Overhead （并行开销）
- Scalability （可扩展性）

存储器架构

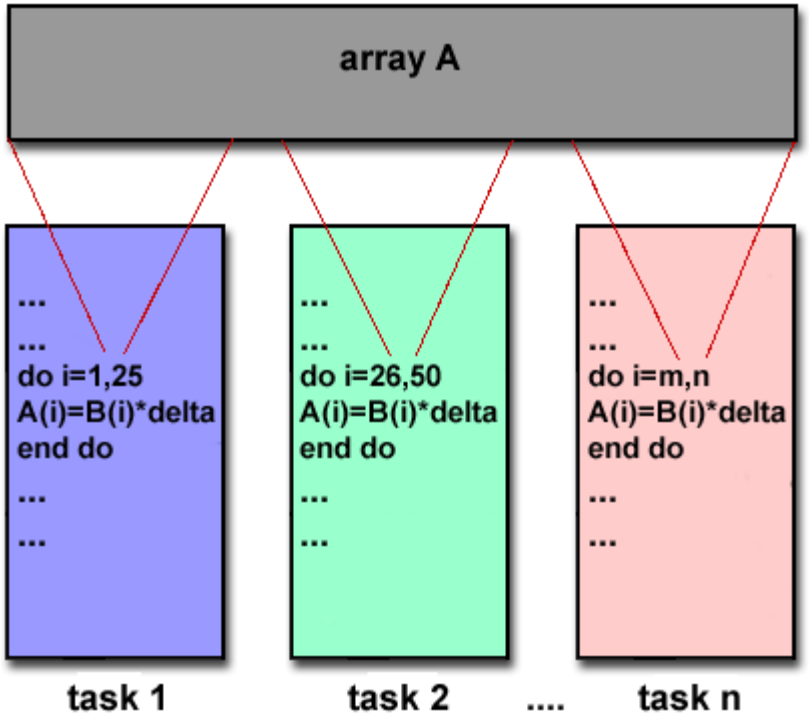
- Shared Memory
- Distributed Memory
- Hybrid Distributed-Shared Memory

并行编程模型

- 共享存储模型Shared Memory Model
- 线程模型Threads Model
- 消息传递模型Message Passing Model
- 数据并行模型Data Parallel Model

具体实例

- OpenMP
- MPI
- Single Program Multiple Data
- (SPMD)
- Multiple Program Multiple Data
- (MPMD)

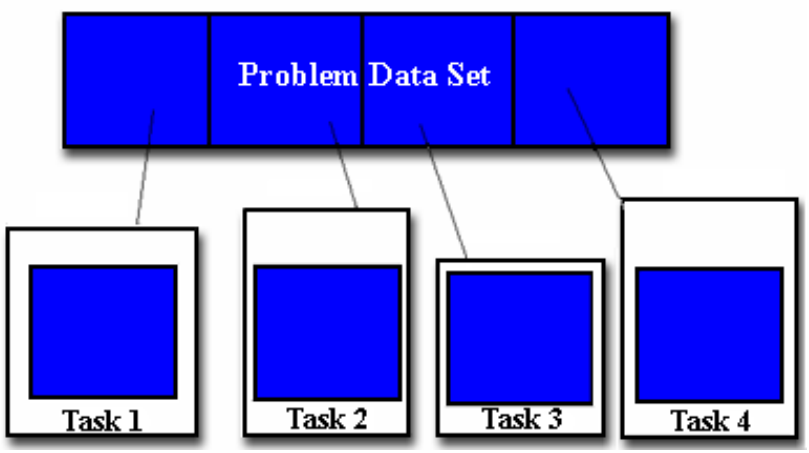


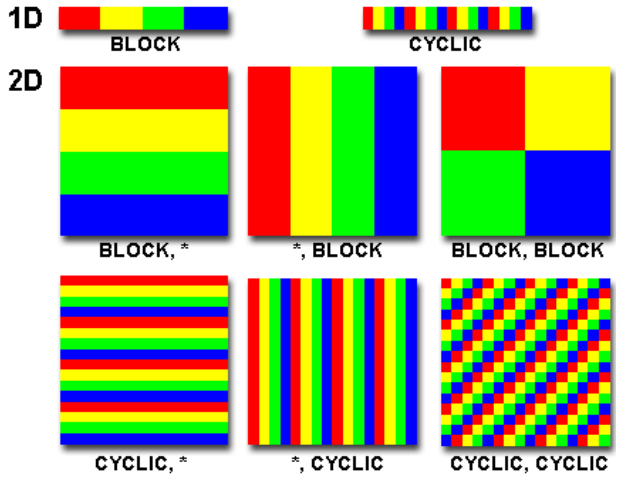
设计并行处理程序和系统

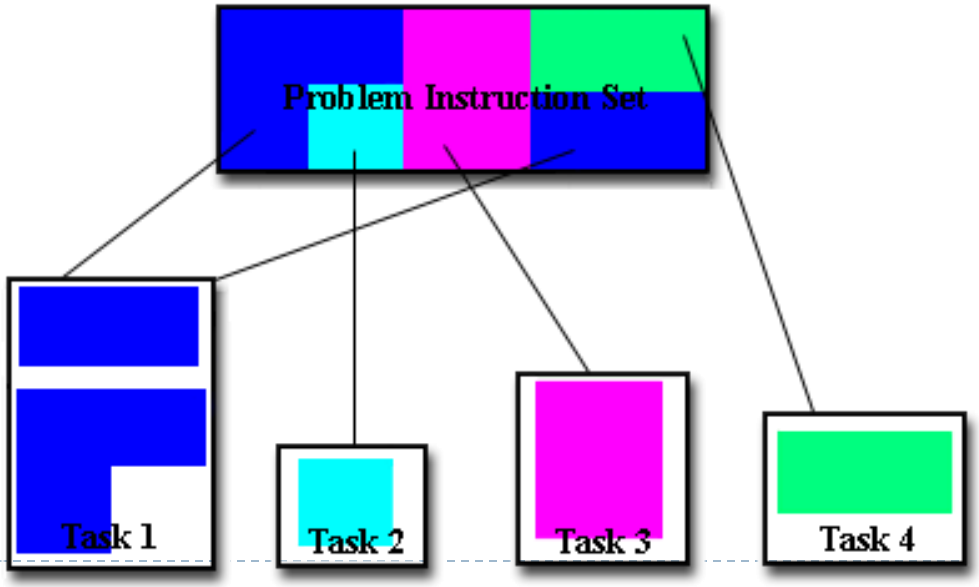
- 自动和手动并行
- 理解问题和程序
- 分块分割
- 通信
- 同步
- 数据依赖

- 负载均衡
- 粒度
- I/O
- 成本
- 性能分析和优化

数据和任务分割

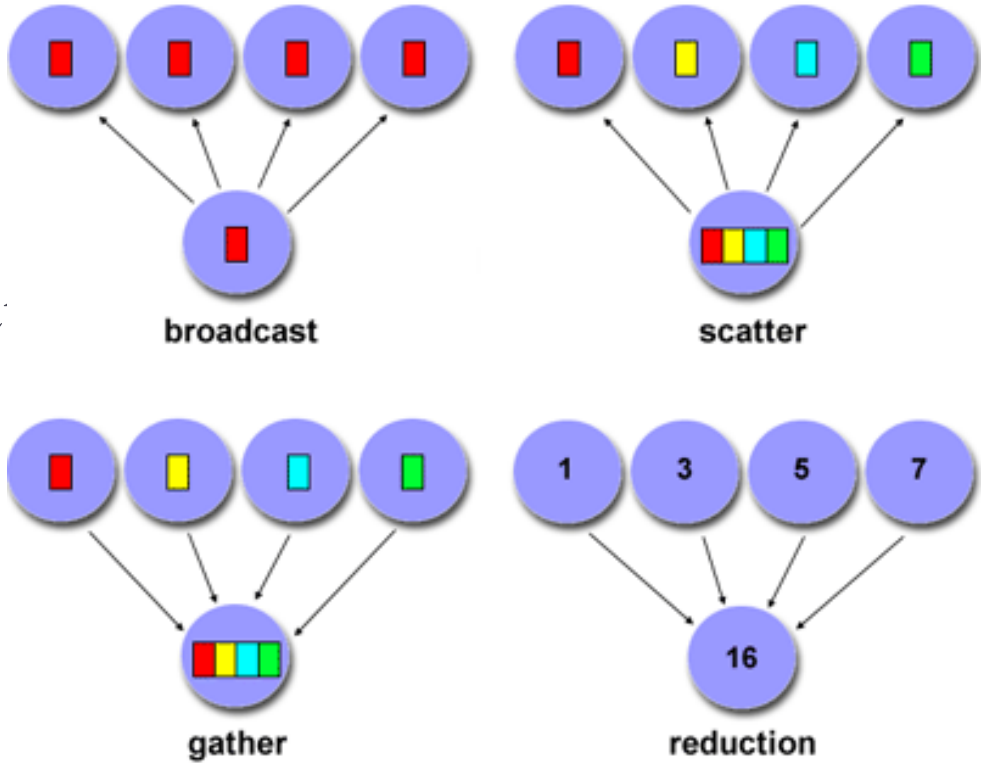

Problem Data Set
Task 1
Task 2
Task 3
Task 4


1D
BLOCK
CYCLIC
2D
BLOCK, *
*, BLOCK
BLOCK, BLOCK
CYCLIC, *
*, CYCLIC
CYCLIC, CYCLIC


Problem Instruction Set
Task 1
Task 2
Task 3
Task 4

通信和同步

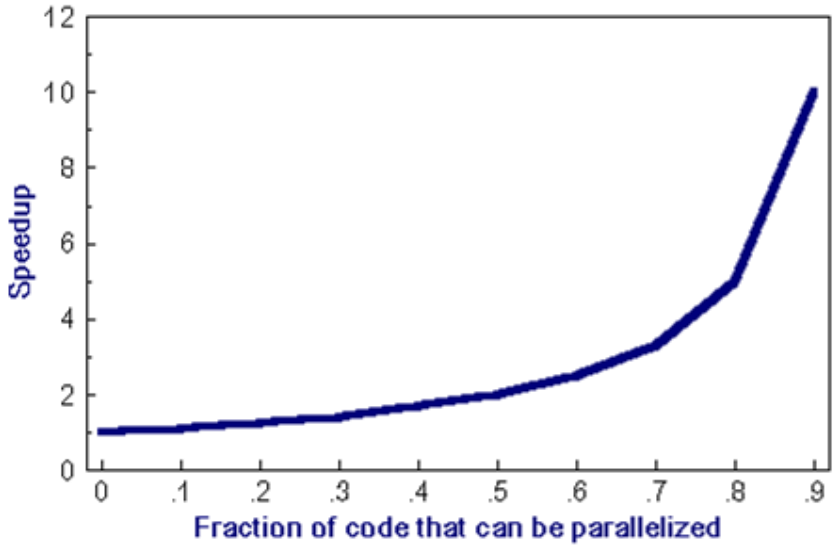
- 同步：
  - Barrier
  - Lock / semaphore
  - Synchronous communica



Amdahl's Law

- Amdahl's Law 程序可能的加速比取决于可以被并行化的部分。

$$\text{speedup} = \frac{1}{1 - P}$$



- 如果没有可以并行化的, P = 0 and the speedup = 1 (no speedup).如果全部都可以并行化, P = 1 and the speedup is infinite (in theory).
- 如果50% 可以并行化, maximum speedup = 2,

Amdahl's Law

- 如果有N个处理器并行处理

$$\text{speedup} = \frac{1}{\frac{P}{N} + S}$$

- P = 并行部分, N = 处理器数量 and S = 串行部分

Amdahl's Law

- 并行化的可扩展性有极限. For example, at P = .50, .90 and .99 (50%, 90% and 99% of the code is parallelizable)

	speedup		
N	P = .50	P = .90	P = .99
10	1.82	5.26	9.17
100	1.98	9.17	50.25
1000	1.99	9.91	90.99
10000	1.99	9.91	99.02