

上海超级计算中心

Shanghai Supercomputer Center

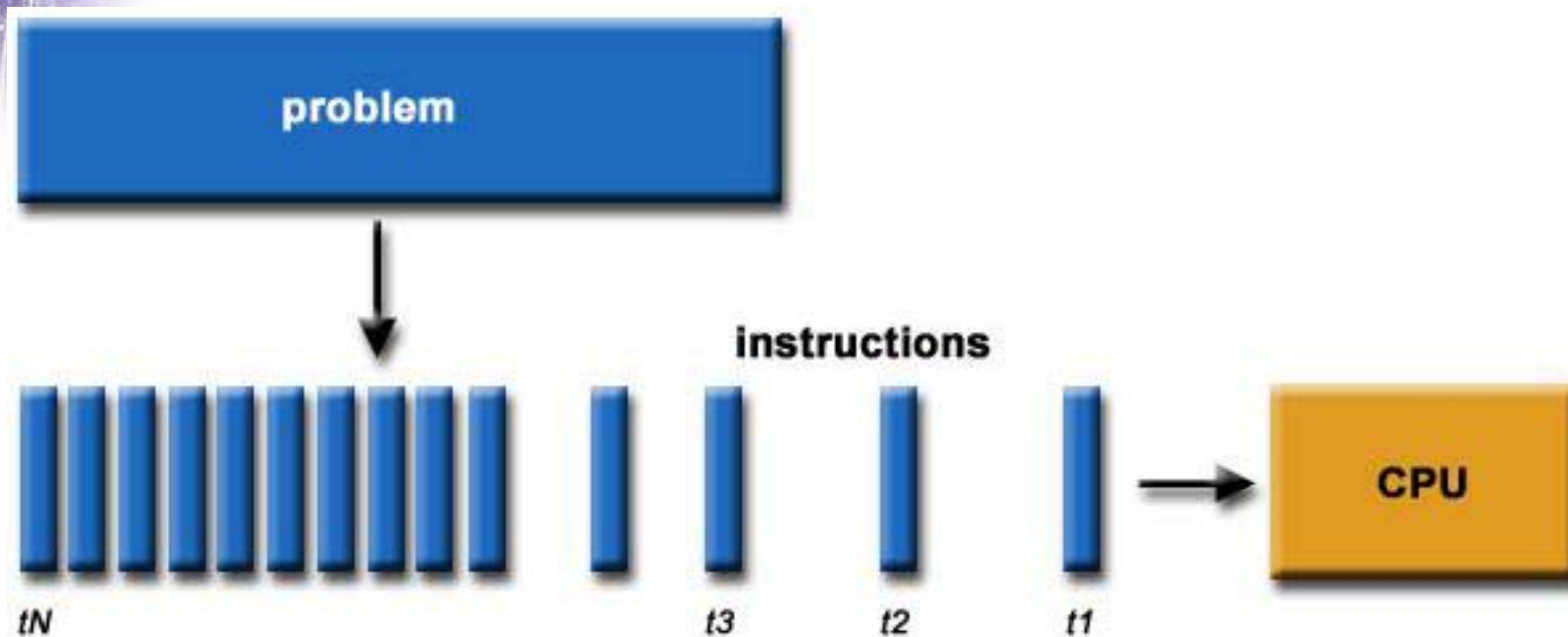


并行计算的概述

上海超级计算中心



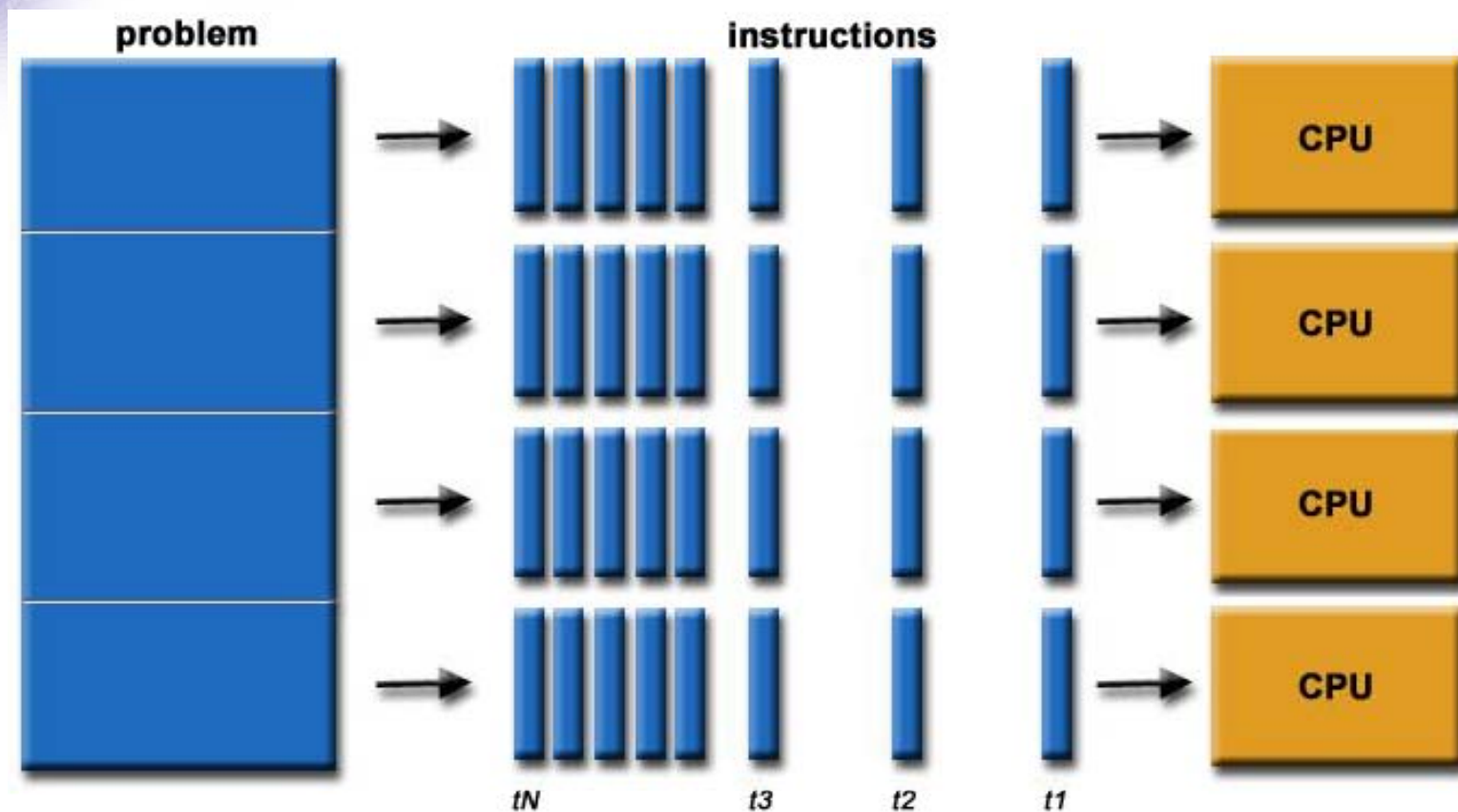
什么是并行计算?



- 运行于单个处理单元
- 顺序执行
- 一次执行一条指令



什么是并行计算？



什么是超级计算？

- 超级计算机 Supercomputer
 - 当前处理能力最强、运算速度最快的一类计算机
- 超级计算 Supercomputing
 - 利用超级计算机所进行的计算，通常为了解决普通计算机所不能完成的大型复杂问题



什么是并行计算？

- 相对于串行计算
- 简单说，是指利用多个部件共同完成计算任务



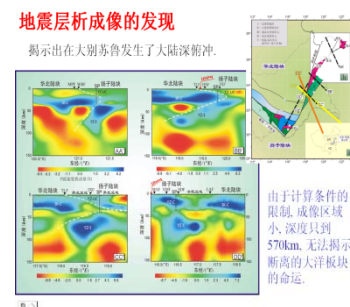
A parallel computer is a collection of processing elements that cooperate to solve large problems fast

David E. Culler U.C. Berkeley

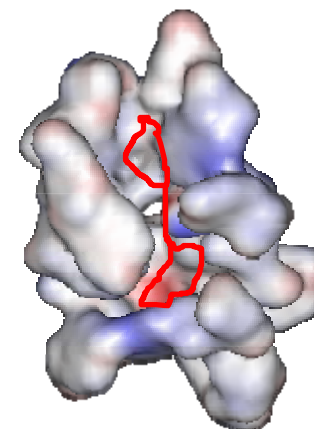


为什么要并行计算？ ——大量应用需求的驱动

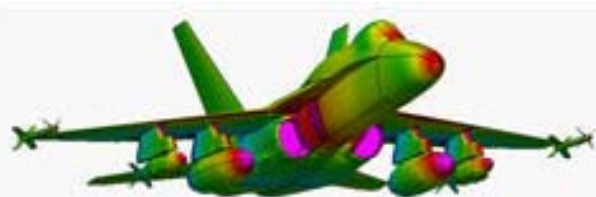
- 有限时间内解决复杂计算问题
 - 汽车碰撞实验：32 CPU 4小时（一个工况）
 - 中尺度天气预报：128 CPU 4小时，每天定时
 - 药物筛选：64 CPU，500万化合物，一年
 - 蛋白质折叠，256CPU、2个月只能算一个纳秒过程
 - 宇宙大尺度结构模拟：256 CPU，6个月



地震层析成像



药物筛选



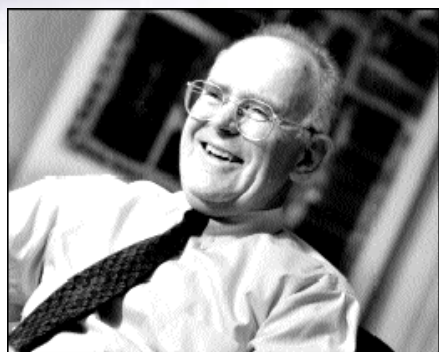
飞机设计



汽车碰撞

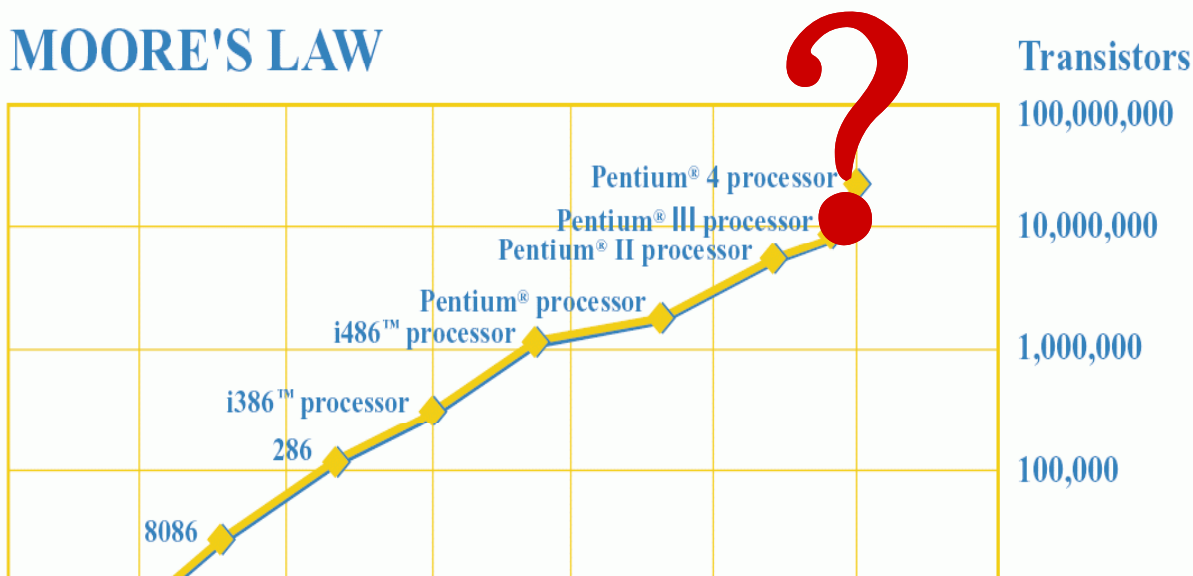


为什么要并行计算？ ——计算机技术发展的推动



芯片上晶体管数目
每18个月增长一倍

MOORE'S LAW

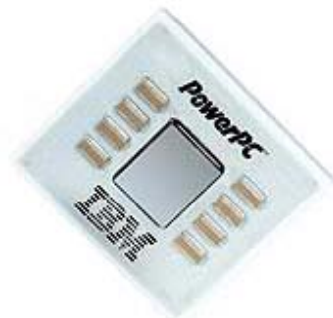
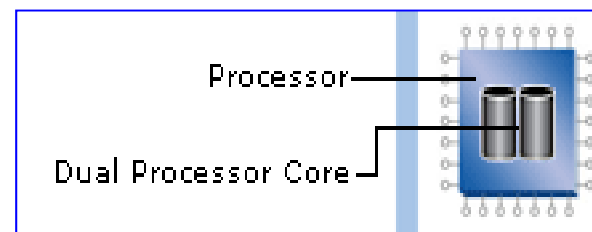
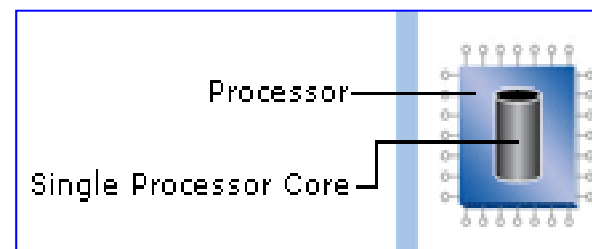


- 工艺：90nm→65nm→32nm，逼近极限
- 主频：从3.0GHz到4.0GHz，步履艰难
- 功耗：单芯片百瓦功耗，mini型“核反应堆”



为什么要并行计算？ ——多核CPU的发展

- 在单个芯片上内置多个处理单元——“核”
- 每个处理器视为小型的并行计算机
- 双核 → 多核
- **并行计算已经成为必然！**





并行计算的大量应用（传统应用）

- 气候环境：天气预报、气候、气象等
- 生物技术：基因、蛋白质、药物设计
- 能源：石油勘探与模拟、核能模拟
- 材料：纳米材料、高分子材料设计
- 制造：外形设计、碰撞试验、发动机燃烧
- 国防和国家安全：密码破译、先进武器制造
- 海量数据处理
-



并行计算怎么算？

4 并行应用开发



3 并行算法设计



2 并行计算机系统软件



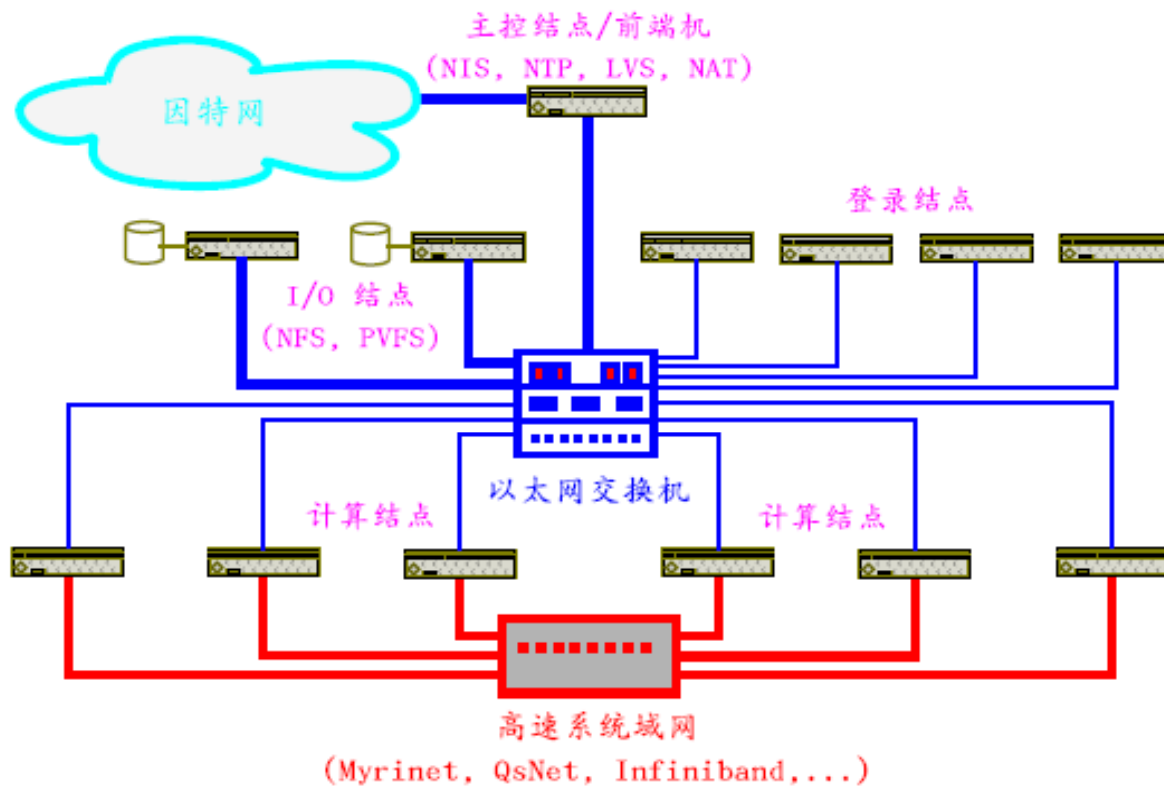
1 并行计算机体系结构



并行算法	并行应用
并行通讯协议和实现	
操作系统	
网络设备和互联	
处理器和硬件平台	



典型的机群系统结构



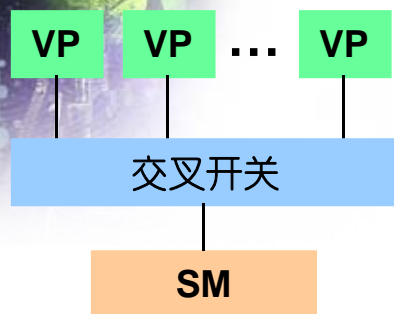


主要并行计算机类型

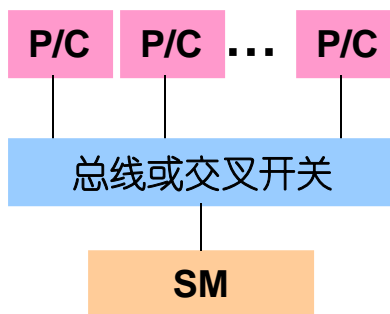
- 并行向量机
 - Parallel Vector Processor, 银河1号, 地球模拟器
- 对称多处理机
 - Symmetric Multiprocessor, 曙光1号, SGI Power Challenge
- 大规模并行处理机
 - Massively Parallel Processor, 神威I, IBM SP2
- 分布共享存储多处理机
 - Distributed Shared Memory, SGI Origin2000/3000
- 集群或集群系统
 - Cluster, Cluster of Workstation, 曙光4000A, 深腾6800



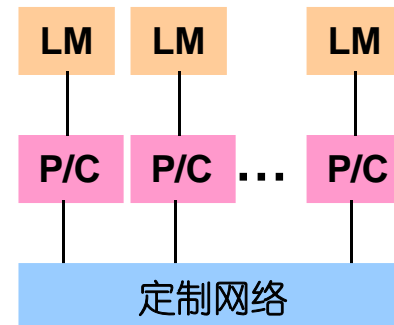
并行计算机体系架构



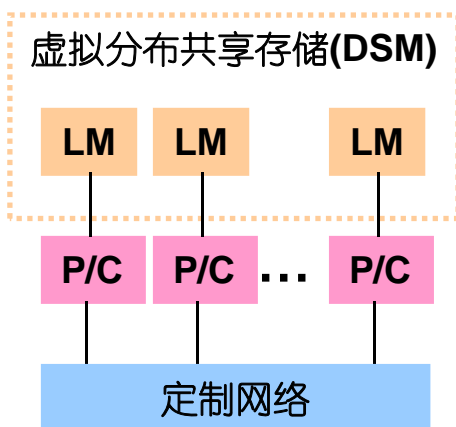
(a) PVP



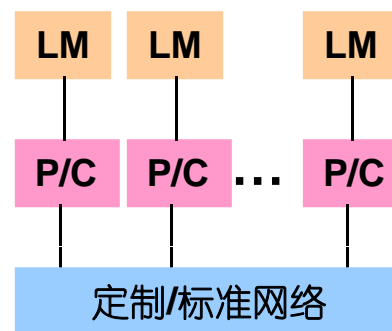
(b) SMP, 物理上单一地址空间



(c) MPP, 物理/逻辑上多地址空间



(d) DSM 逻辑上单一地址空间

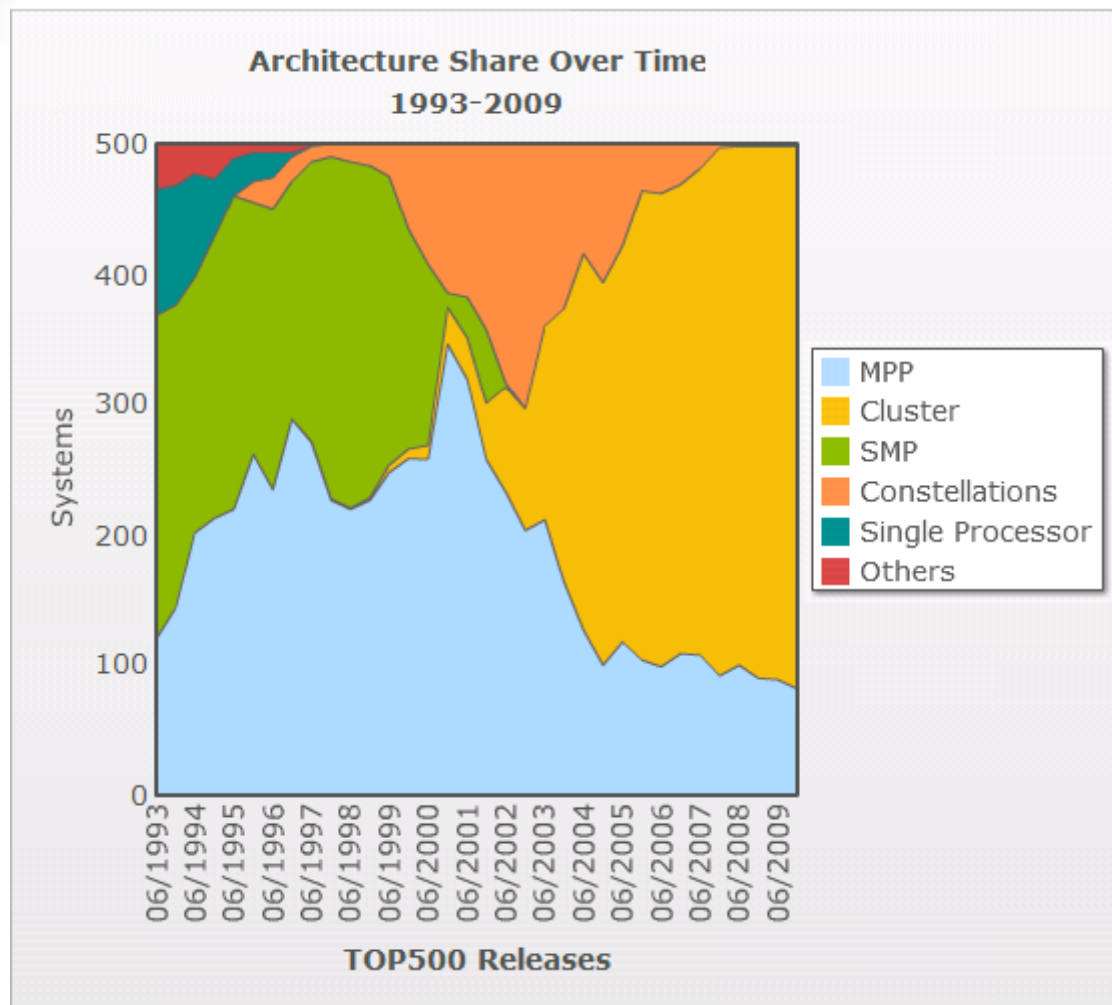


(e) Cluster/COW, 物理/逻辑上多地址空间

体系架构理论模型: **UMA, NUMA, cc-NUMA**



Top500 体系架构趋势 (1993-2009)





当前世界主要超级计算机

- TOP500, 2010年11月

Rank	Site	Computer	Architecture
1	National Supercomputing Center in Tianjin, China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C NUDT	MPP
2	DOE/SC/Oak Ridge National Laboratory, United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz Cray Inc.	MPP
3	National Supercomputing Centre in Shenzhen ,(NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU Dawning	Cluster
4	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows NEC/HP	Cluster
5	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 12-core 2.1 GHz Cray Inc.	MPP



提 纲

- 前言
- 并行计算机
- 并行计算方法
- 并行软件开发
- 展望与挑战



寻找并行性

- **数据并行**: 分解数据, 并行处理

```
for i=0 to 99 do  
  a[i]=b[i]+c[i]  
end for
```

- **功能并行**: 分解问题, 并行处理

```
a=2 b=3  
Task1 = (a+b)/2  
Task2 = a*b  
Result = Task1+Task2
```

- **流水线**: 依据次序, 时域并行



并行算法设计三个要点

- 寻求问题求解过程中的并行性
- 寻求并行算法与并行计算机结构的最佳匹配
- 合理的组织任务，减少额外开销



并行算法设计的四个步骤

1. 划分

- 数据分解或功能分解

2. 通信

- 确定通信模式(局部/全局),负载均衡,

3. 聚集

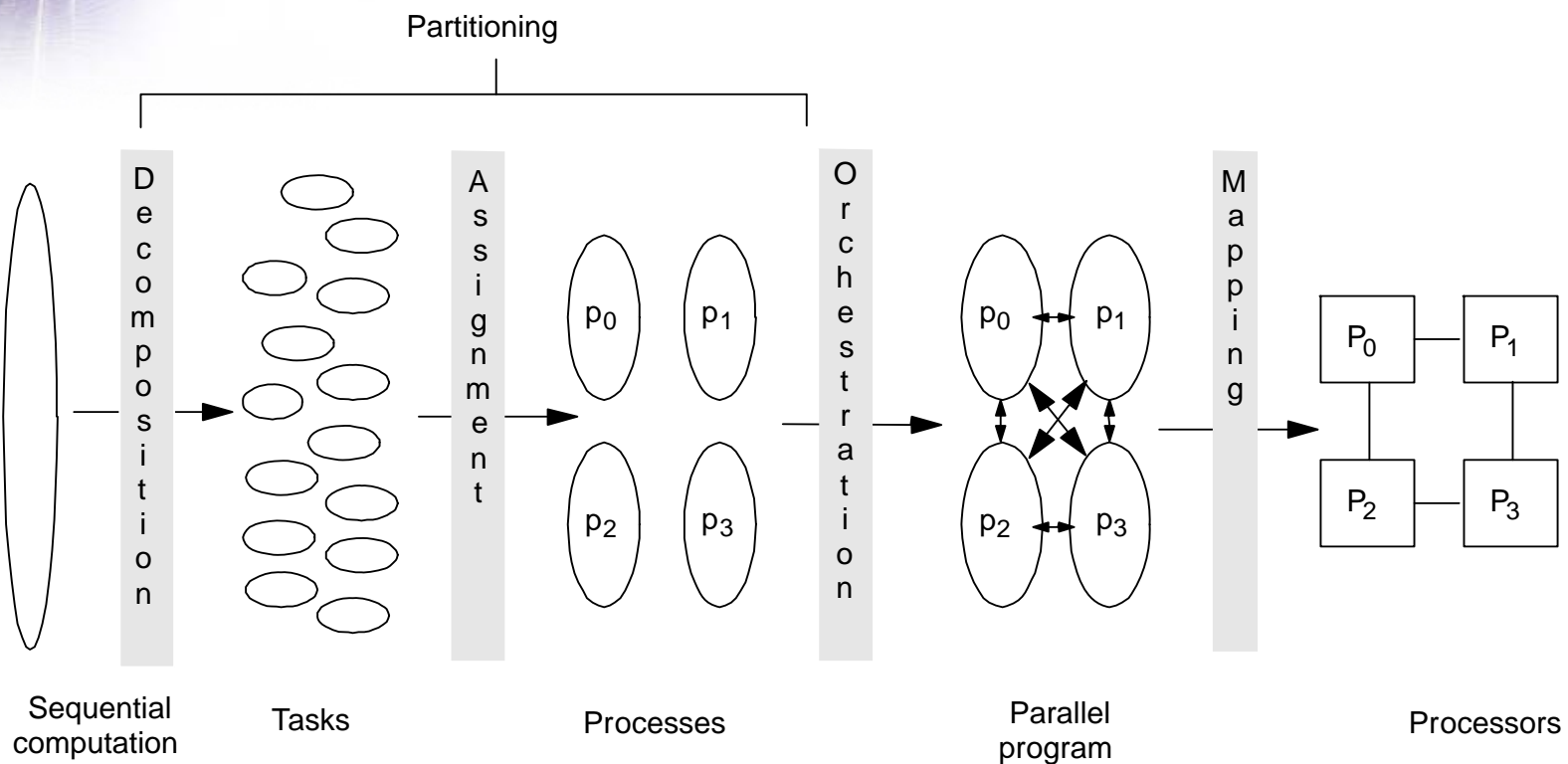
- 依据计算机规模聚合并行化,最小化并行开销,可扩展性

4. 映射

- 将任务分配给处理器



并行算法设计的四个步骤





Partition 分解

- 分解，即将大规模的计算量分解成小份task
 - 区域分解 domain decomposition
 - 任务分解 functional decomposition
- 注意事项 check list
 - task数至少比并行处理器数目多
 - 每个task对应的计算量相当
 - 分解task数目要和问题规模相对应
 - 分解需要尽可能的避免不必要的通讯/存储



Communication 通讯

- 每个子task可以同时(concurrency)在不同处理器上计算，但是无法避免子task间交换数据
 - 区域分解: task间通讯比较难分析
 - 任务分解: task间通讯直接，即task的数据流
- Communication注意事项 check list
 - 子task间通讯平衡
 - 任务间的全局通讯
 - 计算和通讯尽量能够重叠(overlap)
 - 不同task的计算能同时(concurrently)进行，



- Communication注意事项 check list
 - 各子task间通讯平衡
 - 任务间的全局通讯可能会降低并行算法的可扩展性，稳定性
 - 计算和通讯尽量能够重叠(overlap)，如果不，需要用 divide and conquer分解算法
 - 不同task的计算能同时(concurrently)进行，



Agglomeration (聚集)

- 将分解和通讯得到的算法具体化
 - 将子task聚集在一起，增加子任务大小，减少总的子任务数目
 - 对计算和数据的重复存放
- 在聚集阶段，需要面对3个目标
 - 减少通讯开销
 - 增加每个子任务的粒度granularity
 - 一定的灵活性，保证映射阶段的可扩展性
 - 减少软件工程的开销

Mapping (映射)

- 建立task和并行处理器的映射关系
- 目标：减少总的执行时间
 - 增强concurrency
 - 将相互通讯密集的task放在临近的节点上
- 映射问题有大量理论研究 (NP-complete)
- 针对需要负载不均衡(load imbalance)
 - 考虑动态负载平衡
 - 需要增加额外的通讯/计算开销



不同的并行层次

专业领域科学家

问题本身

应用层

专业领域科学家

分析建模

方法层

计算数学家

并行算法

算法层

计算机专家

并行实现

程序层

并行粒度



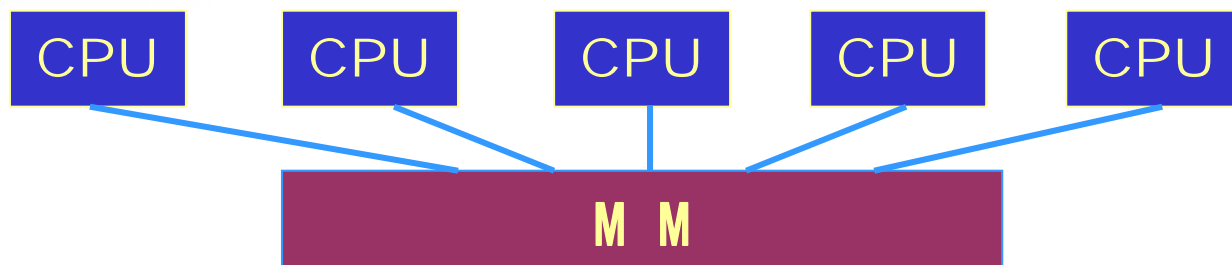
提 纲

- 前言
- 并行计算机
- 并行计算方法
- 并行软件开发
- 展望与挑战

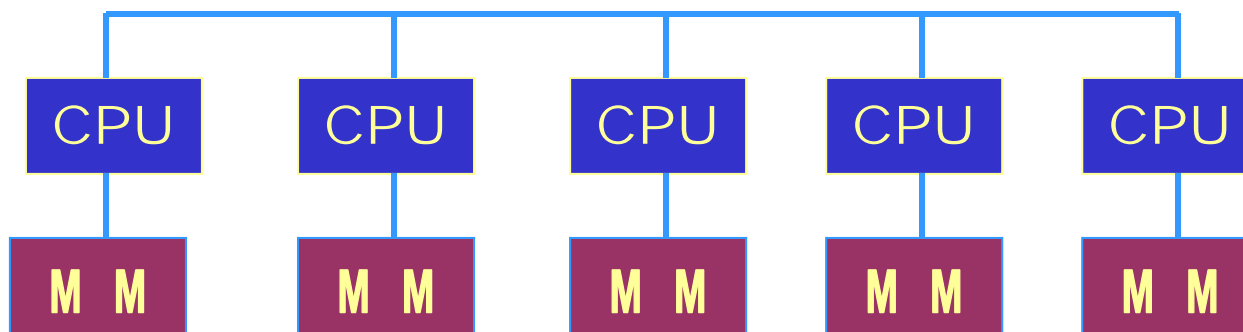


两种内存模型

- 共享存储



- 分布式存储





并行程序实现技术

- 共享存储
 - 自动并行化
 - 线程并行
 - 编译器制导语言，OpenMP
- 分布式存储—消息传递
 - MPI, Message Passing Interface
 - PVM, Parallel Virtual Machine
- 其他
 - 数据并行，HPF
 - 混合同行



程序并行化途径

- 串行程序自动并行化
 - 编译器执行自动并行化，只对部分应用有效
 - 目前为止没有行之有效的方法和工具
 - 半自动化并行途径(编译制导)成为目前的一个重要途径
- 使用全新的并行语言，如HPF
 - 需要重写已有程序
- 串行程序+并行化扩展
 - 基于已有程序进行改写
 - 当前主流的并行程序实现途径
 - MPI, OpenMP



谢谢，欢迎提问和讨论！