

并行处理与体系结构 (1)

Parallel Processing and Architecture

2025-3-23

1

第一课 开场白

- ▶ 课程概况
- ▶ 课程入门基础知识

▶ 2

2

课程概况

- ▶ **课程名称：**并行处理与体系结构(Parallel Processing and Architecture)
 - ▶ 并行计算机体系结构前沿讲座→高等并行计算机体系结构→
 - ▶ 研究生专业学位类别核心课
 - ▶ 48学时，3学分；课表安排4-15周，每周日1-4节，4-9周北京学院路校区新主楼A208，10-15周杭州国新院校区教学一号楼2002
- ▶ **主讲教师**
 - ▶ 栾钟治
- ▶ **联系信息**
 - ▶ 中德所
 - ▶ 办公地点：新主楼G413
 - ▶ 联系方式：
 - ▶ email: luan.zhongzhi@buaa.edu.cn
 - ▶ tel: 13661355707
 - ▶ qq:9281971
 - ▶ 微信: rickluan

▶ 3

3

课程概况（续）

- ▶ **教学方式**
 - ▶ 课堂讲授结合学生相关论文阅读、研讨和总结
 - ▶ 学生完成分组报告
- ▶ **考核方式**
 - ▶ 分组报告 60%
 - ▶ 期末分组课程实践 40%

▶ 4

4

2021年研究生课程卓越教学奖



▶ 5

5

课程概况（续）

▶ 参考教材

参考书 Reference	作者 Author	出版社 Press
Readings in Computer Architecture	Mark D. Hill etc.	Morgan Kaufmann,
Computer Architecture: A Quantitative Approach	John L. Hennessy, A. Patterson	Morgan Kaufmann,
多核并行体系结构基础	[美] 汤孟岩 著 钱德沛、杨海龙、王锐、乐钟治、刘铁 译	机械工业出版社

▶ 6

6

课程目标和内容

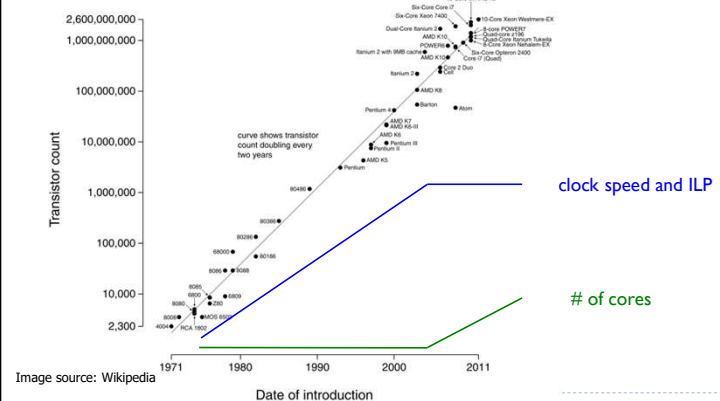
- ▶ 随着并行体系结构逐渐成为主流，可编程性、可扩展性以及功耗等等成为急需解决和研究热点问题
- ▶ 以共享内存并行计算机为对象，从不同的角度探讨并行体系结构的设计问题，让学生对并行体系结构的基本概念和核心问题有较为深入的认识，给今后的研究和实践打好基础
 - ▶ 现代共享内存并行计算机体系结构
 - ▶ 并行计算系统的软硬件协同设计
 - ▶ 并行计算机系统性能分析
 - ▶

▶ 7

7

串行性能的终结

Microprocessor Transistor Counts 1971-2011 & Moore's Law



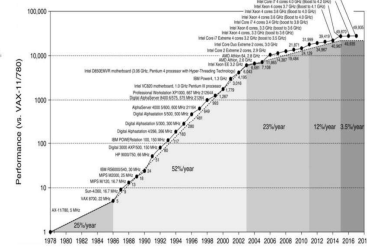
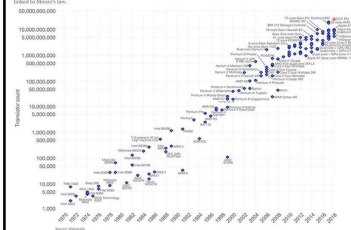
▶ 8

8

Moore's Law

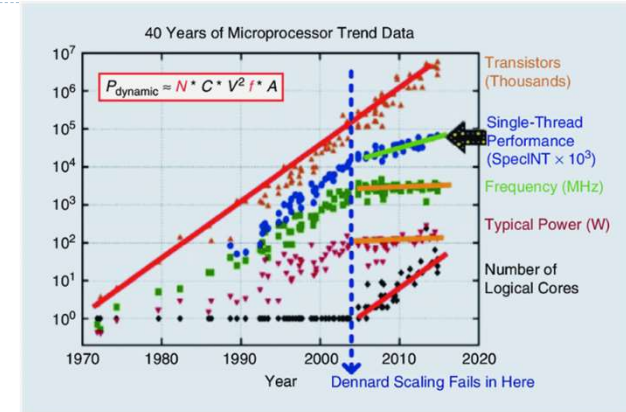
Moore's Law - The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This observation is important to other aspects of technological progress - such as processing speed or the price of electronic products - via Silico's Moore's Law.



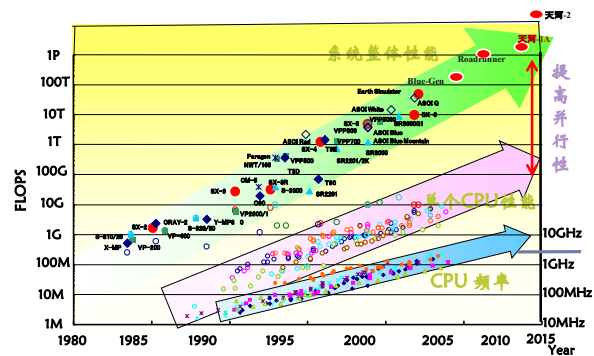
9

Dennard Scaling Law



10

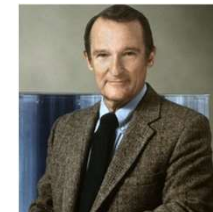
并行是提高性能（达到“超级”）的必由之路



11

第一台“超级计算机” & 超级计算之父

CDC6600, 1964年发布

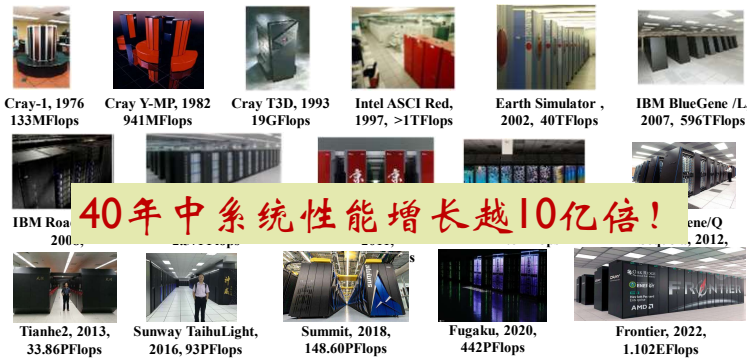


Seymour Cray, 西摩·克雷 (1925-1996)

- ~3MFlops(40MHz主频)
- IBM 7030(Stretch) 性能的10倍
- 售价800万美元, 卖出大约150台

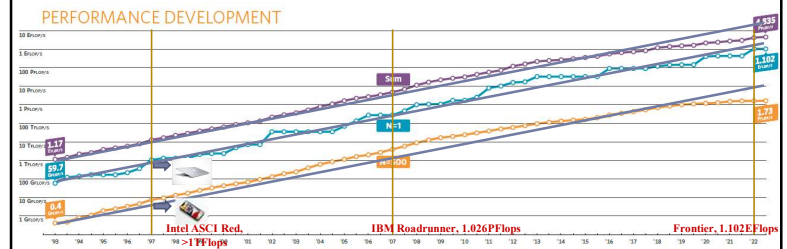
12

超级计算机发展之路



13

TOP500揭示超级计算发展趋势



千倍“定律”：超级计算机的速度每十年提升千倍

14

IBM超级计算机蓝色基因/Q



Blue Gene/Q Sequoia, 峰值性能 20.1PFlops
16-core chip
1.6 million cores
2012 No.1

15

15

天河2号



Tianhe-2, 峰值性能 100PFlops
12-core Intel Xeon 2.2GHz + Xeon Phi (Matrix 2000)
4.98 million cores
2013 No.1

16

16

神威·太湖之光



Sunway TaihuLight, 峰值性能 125.4PFlops
260-core SW26010 1.45GHz
10.65 million cores
2016 No.1

▶ 17

17

Summit



Summit, 峰值性能 200.8PFlops
IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100
2.41 million cores
2018 No.1

▶ 18

18

Fugaku (富岳)



Fugaku, 峰值性能 513.85PFlops
ARM A64FX 48C 2.2GHz
7.3 million cores
2020 No.1

▶ 19

19

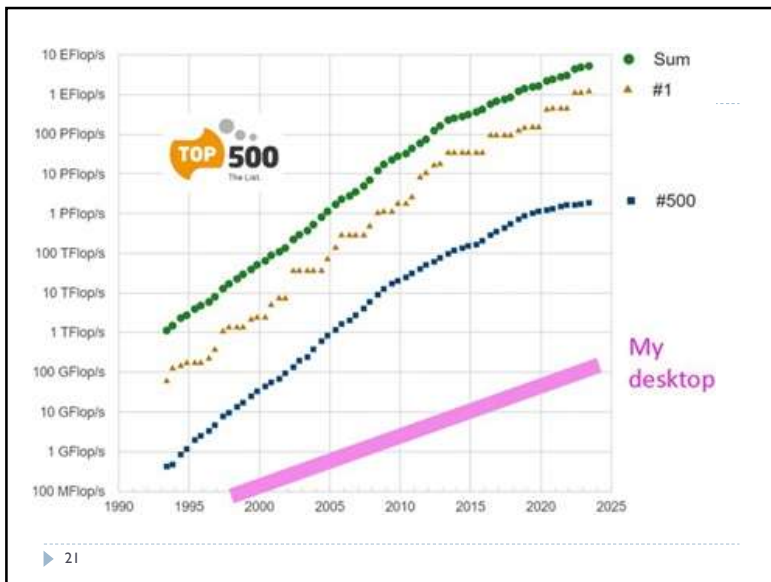
Frontier



Frontier, 峰值性能 1.102 Exaflops
AMD Milan "Trento" 7A53 Epyc 64C CPU+
AMD Instinct MI250X GPU
8.73 million cores
2022 No.1

▶ 20

20

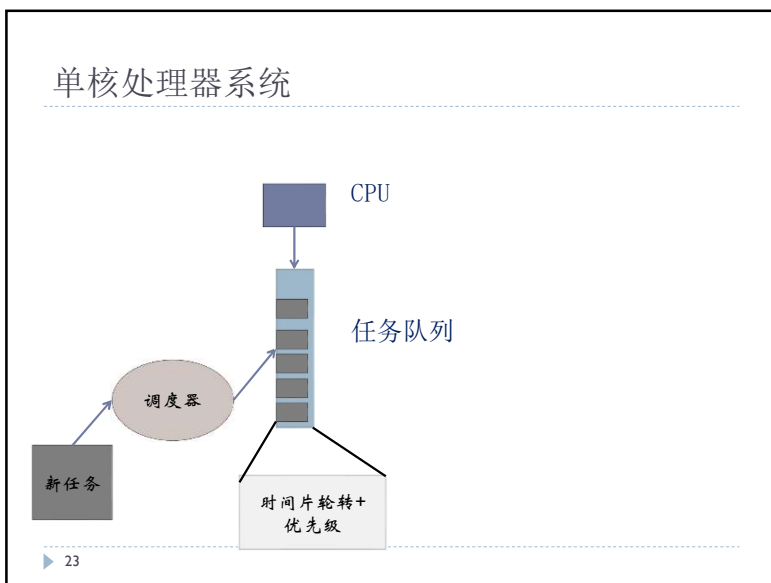


21

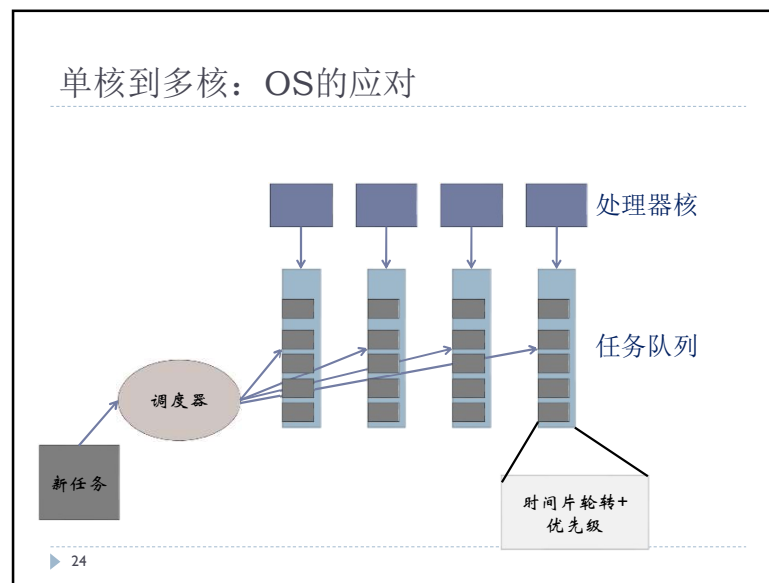
多核处理器让你的电脑变快了吗?

22

22

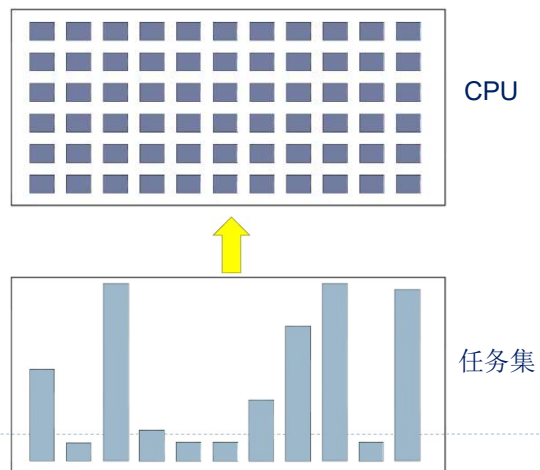


23



24

任务数小于处理器核数？



25

25

软件管理与软件工程译丛

一书在软件领域绝无仅有，
20余年畅销不衰的
必读经典

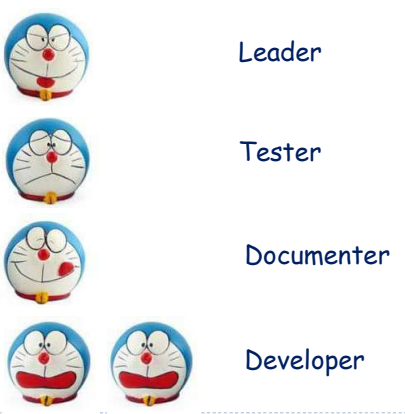


The Mythical Man-Month
人月神话

26

26

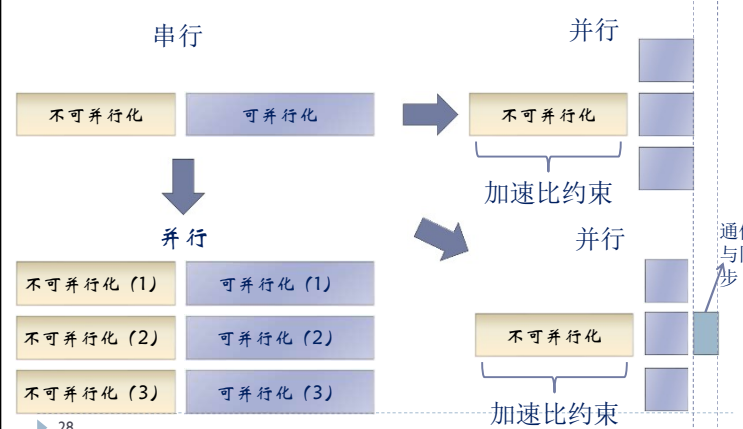
五人开发小组



27

27

Amdahl's Law的神话



28

28

真正不可并行的问题少之又少

29

如何更好的并行化？

剧本编写的并行加速比：



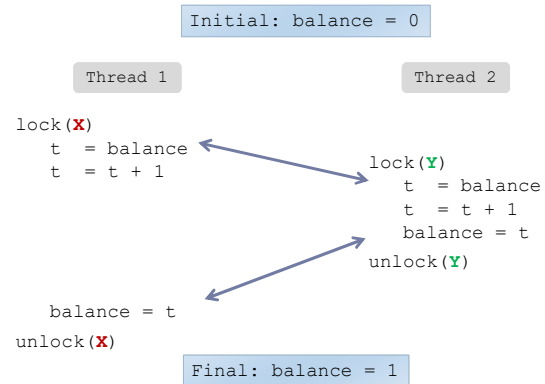
Vs.



人物

30

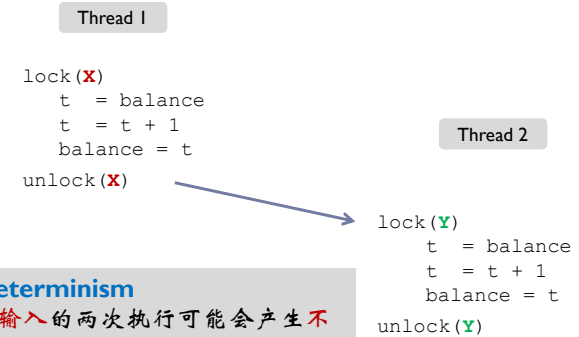
数据竞争（Data-Races）



对balance访问的数据竞争

31

非确定性



Non-determinism

对相同输入的两次执行可能会产生不同的答案

balance = 1 或 2

32

顺序一致性

```

X* x = null;
bool flag = false;

// Producer Thread      // Consumer Thread
A: x = new X();          C: while(!flag);
B: flag = true;          D: x->f++;
    
```

sequential consistency (SC)

[Lamport 1979]

memory operations appear to occur in same global order consistent with the program order

▶ 33

33

能否使程序具有并行的性能和串行的简单性?

▶ 34

34

规则的数据并行

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} = ?$$

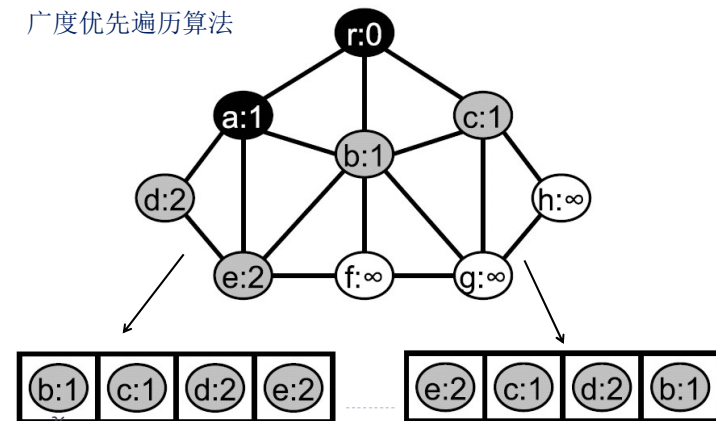
提前确定数据结构，数据间无差别处理

▶ 35

35

不规则程序的并行

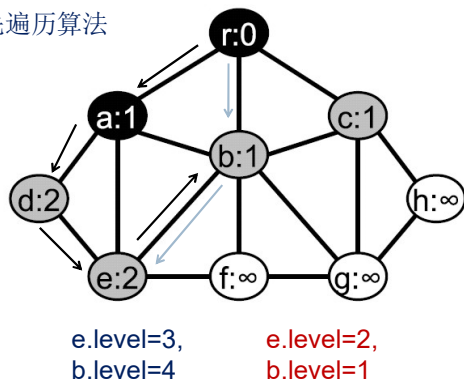
广度优先遍历算法



36

不规则程序的并行

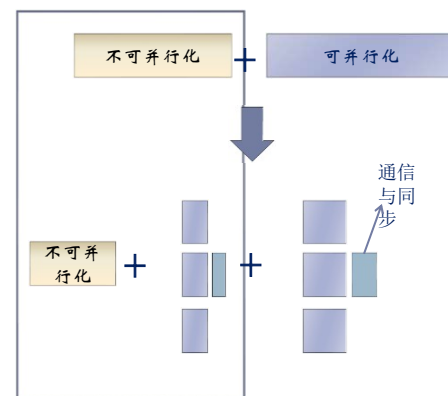
广度优先遍历算法



▶ 37

37

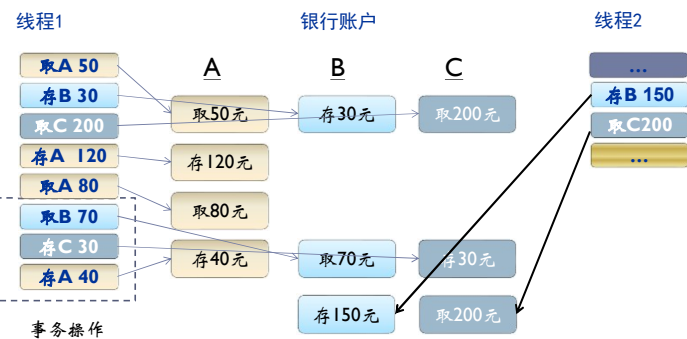
重新审视 Amdahl's law



▶ 38

38

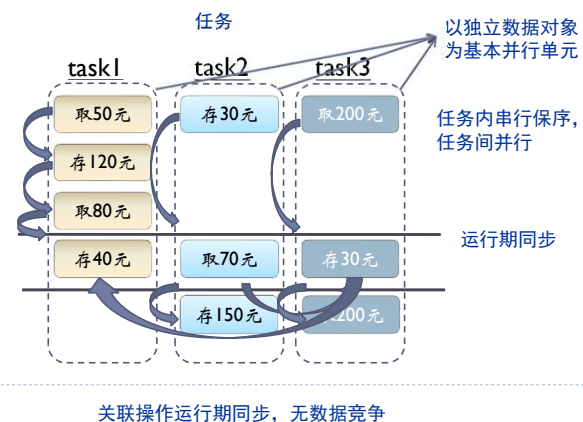
以数据为中心的并行任务分解



▶ 39

39

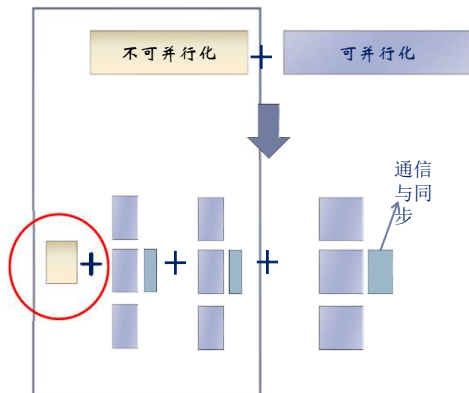
以数据为中心的并行任务分解



▶ 40

40

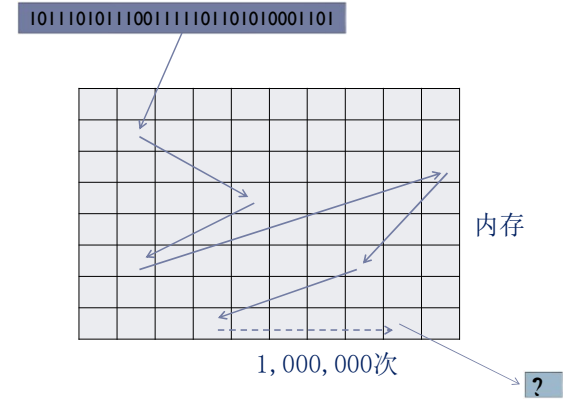
重新审视 Amdahl's law



41

41

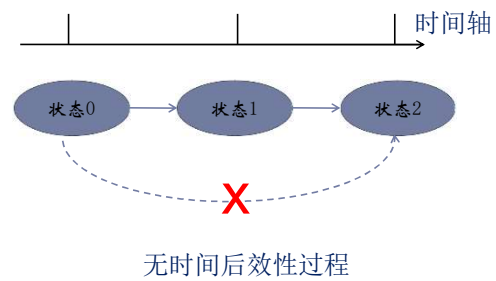
不可并行问题



42

42

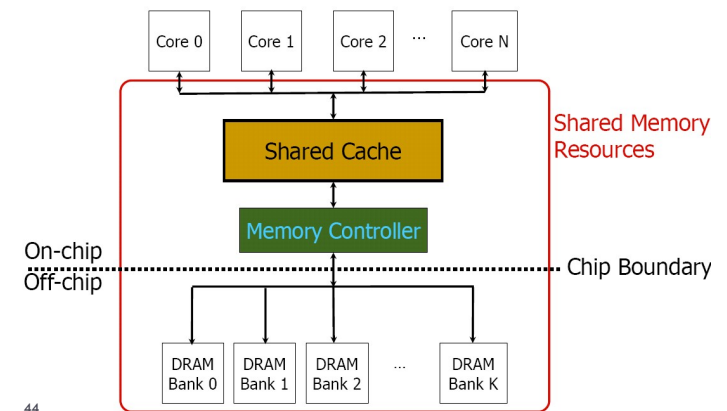
马尔可夫过程



43

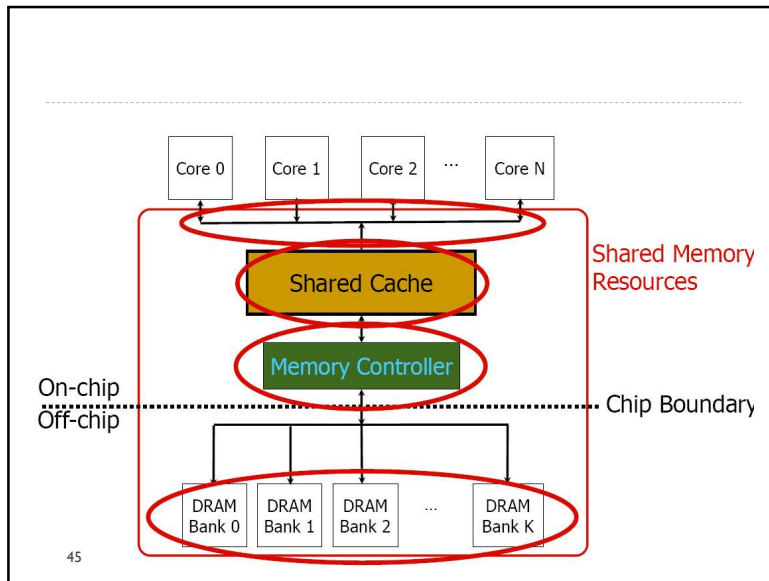
43

硬件角度观察计算机的并行性



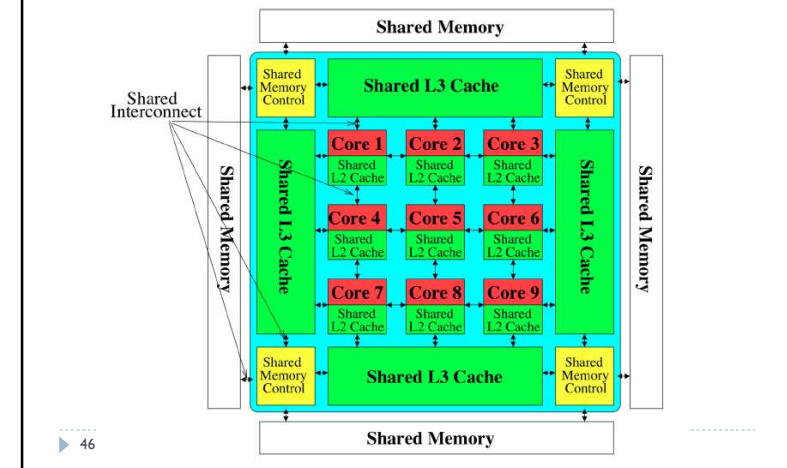
44

44



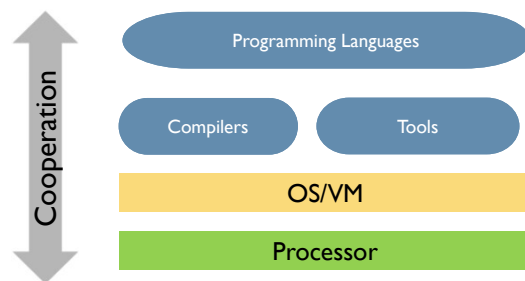
45

换个角度



46

Holistic Solutions-系统性的方法



47

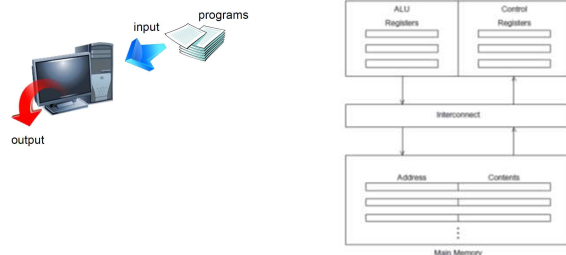
并行结构

- ▶ **并行性**: 同时做多件“事情”
- ▶ “事情”: 指令, 操作, 任务
- ▶ **为什么采用并行结构**
 - ▶ **绝对性能**: 主要目标, 执行时间或任务吞吐量
 - ▶ 程序的执行时间由Amdahl定律控制
 - ▶ **功耗和能耗**
 - ▶ 4N个单元工作在F/4频率下的功耗低于N个单元工作在F频率下
 - ▶ **提高成本效率和可扩展性, 降低复杂性**
 - ▶ 性能价格比
 - ▶ 很难设计出一个复杂单元能够有N个简单单元一样的执行效果
 - ▶ **提高可靠性**: 在空间上冗余执行
- ▶ **关键的使能因素**
 - ▶ 半导体和互连网络技术的发展
 - ▶ 软件技术的发展

48

串行的硬件和软件

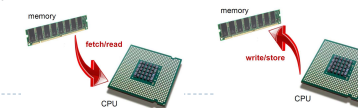
- ▶ 计算机一次只执行一个程序
- ▶ 冯诺依曼结构
 - ▶ 几乎所有计算机都采用的控制流架构
 - ▶ 两个基本特征：存储程序和顺序执行指令



49

关键术语

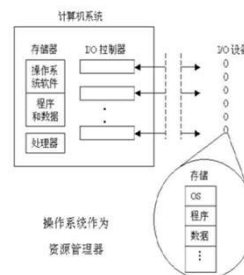
- ▶ 主存储器
 - ▶ 一组位置的集合，每个位置都能够存储指令和数据
 - ▶ 每个位置都由一个用于访问该位置的地址和该位置的内容组成
- ▶ CPU（中央处理器）
 - ▶ 负责决定程序中应该执行哪条指令的控制部件.....BOSS
 - ▶ 负责执行实际指令的算术逻辑单元(ALU).....worker
- ▶ 寄存器
 - ▶ 非常快的存储，是CPU的一部分
- ▶ 程序计数器
 - ▶ 存储下一条要执行的指令的地址
- ▶ 总线
 - ▶ 连接CPU和内存的线和硬件
- ▶ 冯诺依曼瓶颈



50

操作系统“进程”

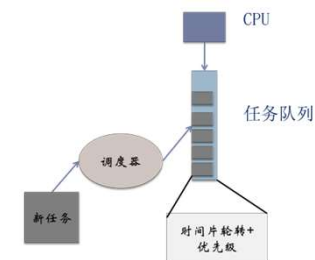
- ▶ 正在执行的**计算机程序的实例**
- ▶ 进程的组成部分
 - ▶ 可执行的机器语言程序
 - ▶ 一块内存空间
 - ▶ 操作系统分配给进程的资源描述符
 - ▶ 安全信息
 - ▶ 关于进程状态的信息



51

多任务处理

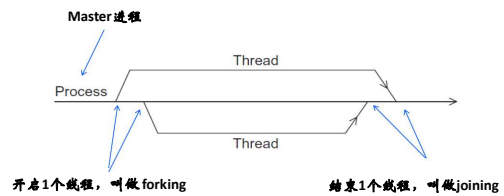
- ▶ 给人一种单处理器系统同时运行多个程序的错觉
- ▶ 每个进程轮流执行(**时间片**)
- ▶ 时间到了之后，会等待直到再次获得时间片(**block**)



52

线程

- ▶ 线程包含在进程中
- ▶ 允许程序员将程序分成(或多或少的)独立的任务
- ▶ 希望当一个线程因为等待资源而阻塞时, 另一个线程有工作要做并可以运行



53

谢谢!

▶ 54

54