性能评价指标

大纲

- 衡量计算机性能的指标
  - 执行时间
  - 吞吐率
- CPU性能公式
- Amdahl 定律：性能优化的基本定律
  - Amdahl 定律适用和不适用的场合

计算机的性能—两个常用的指标

1. 执行时间（Execution Time/Latency）

计算机完成某任务的总时间，包括硬盘访问、内存访问、 CPU执行时间、操作系统开销、和输入/输出活动等所有从开始执行到执行结束的总时间。

性能 = 1 / 执行时间

2. 吞吐率（Throughput）

性能加速比

$$\text{加速比} = \frac{\text{性能}_X}{\text{性能}_Y} = \frac{\frac{1}{\text{性能}_Y}}{\frac{1}{\text{性能}_X}} = \frac{\text{执行时间}_Y}{\text{执行时间}_X}$$

- 系统X用10秒钟执行某个程序，系统Y执行同一个程序花费15秒
  - 系统X 比 系统Y 快 1.5 倍、
  - 系统X 对 系统Y 的加速比（speedup）是 1.5
  - X比Y性能提升了 50%

吞吐率（Throughput）

- **带宽**（bandwidth）：单位时间内完成的任务数量
- **例如：**
  - 多媒体应用
    - 吞吐率高：音频/视频播放流畅
  - Web 服务器：
    - 吞吐率高：在单位时间完成的任务越多越好（管理员角度）
    - 执行时间短：提交的任务越快完成越好（使用者角度）

执行时间与吞吐率

- **互相关联，例如：**
  - 使用更快的处理器
    - 既能缩短一个程序的执行时间，又能提高系统整体的吞吐率
  - 将一个处理器增加为多个处理器
    - 只增加吞吐率，无法缩短单个任务的执行时间

课本练习题

- 假设某个使用桌面客户端和远程服务器的应用受到网络性能的限制，那么对于下列方法，哪个是同时改进了吞吐率和响应时间的？

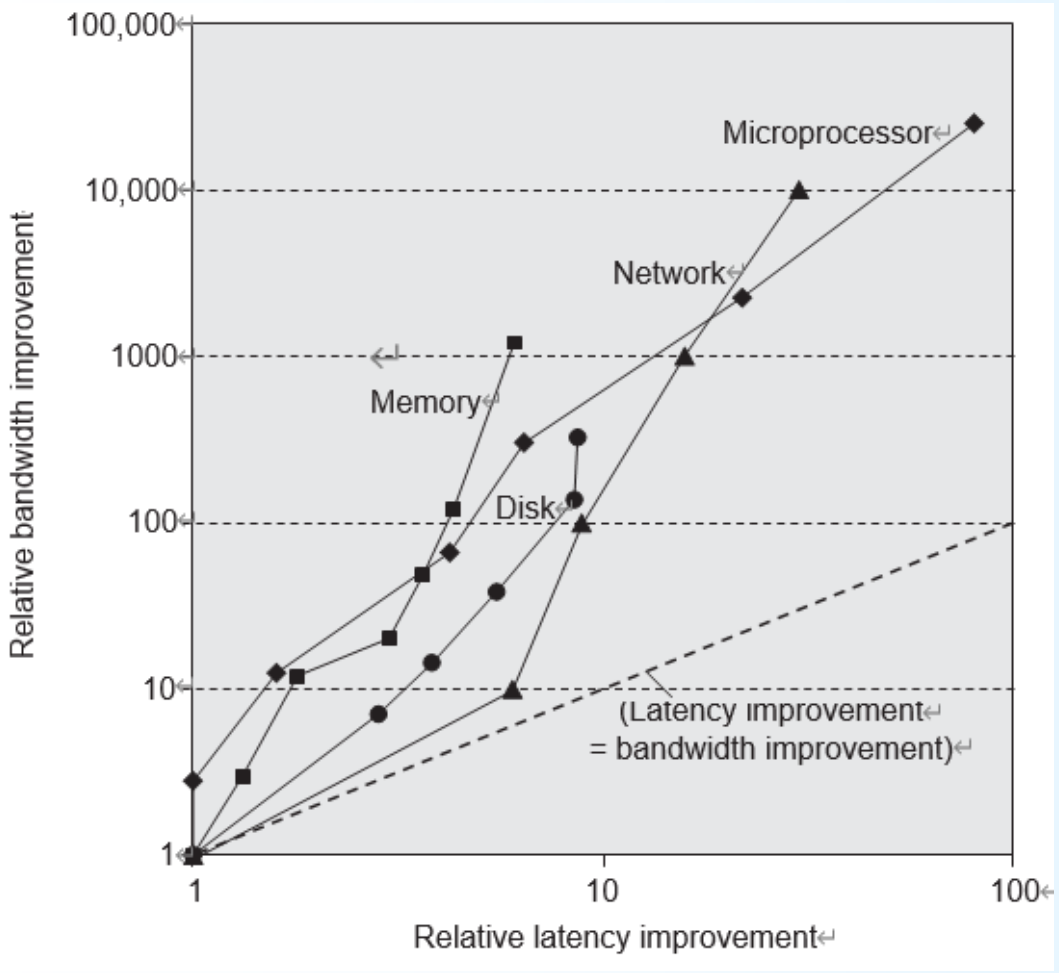
A. 在客户端和服务器之间增加一条额外的网络信道（现在有两条网络信道了）

B. 改进网络软件，从而减少网络通信延迟，但并不增加吞吐率

C. 增加计算机内存

D. 更换运算速度更快的处理器

Performance Trends: Bandwidth over Latency



Log–log plot of bandwidth and latency milestones

Bandwidth / throughput

- 10,000-25,000X improvement for processors
- 300-1200X improvement for memory and disks

Latency / response time

- 30-80X improvement for processors
- 6-8X improvement for memory and disks

执行时间 vs. CPU时间

- 一个程序在输入输出等待时，处理器会切换去执行另外一个程序，以提高系统的运行效率，但这样会延长单个程序的执行时间。
- CPU时间
  - 给定程序任务占用处理器的时间。
  - 更合适用于衡量一个程序在硬件系统中的性能

The CPU Performance Equation

CPU time = IC x CPI x Clock time

where: CPU time = execution time
IC = number of instructions executed (instruction count) 执行指令条数
CPI = number of average clock cycles per instruction
Clock time = duration of processor clock 持续时间 处理器

完成1条指令所要的周期时间

$$\text{CPU time} = \left(\sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock time}$$

where: IC_i = IC for instruction (instruction group) i
CPI_i = CPI for instruction (instruction group) i

CPU执行时间 — 用于性能比较

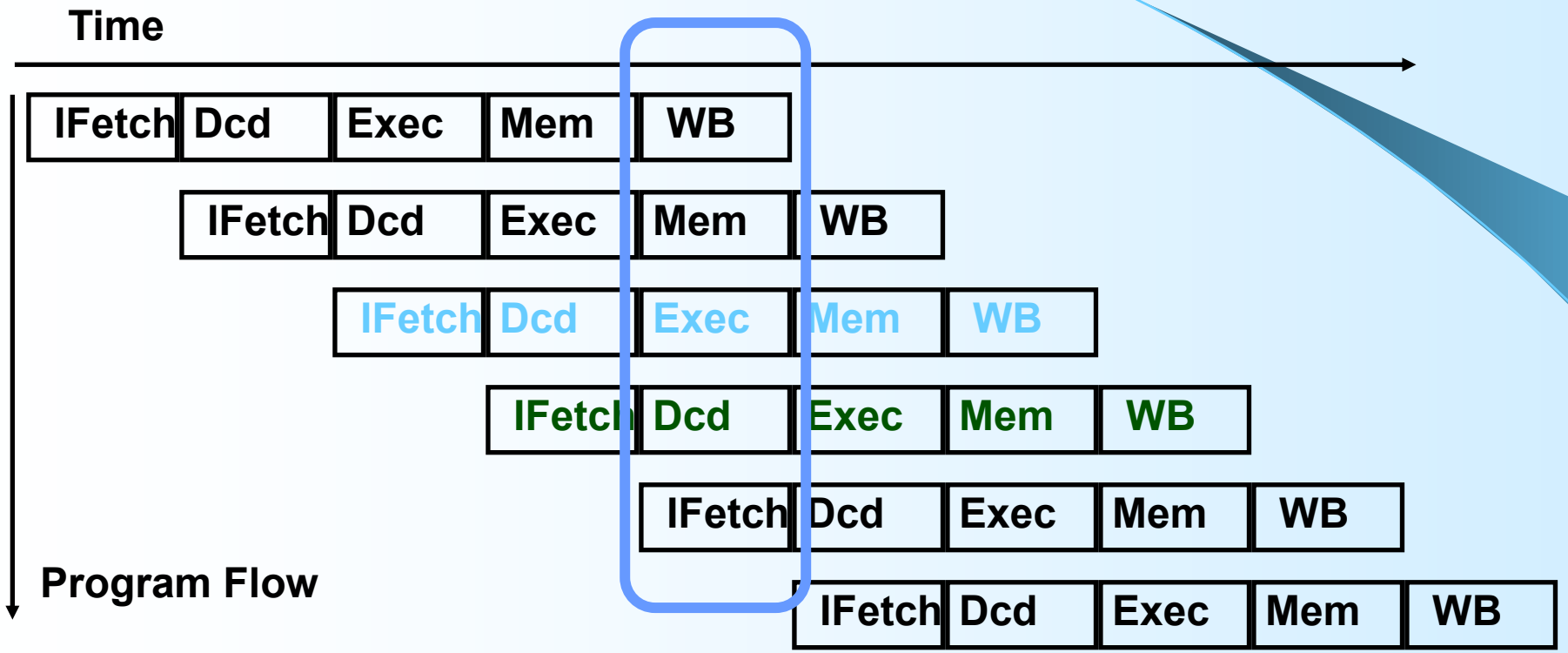
- 例如：
- 假定计算机M1和M2具有相同的指令集体系结构，主频分别为1.5GHz和1.2GHz 。在M1和M2上运行某基准程序P，平均CPI分别为2和1，那么程序在M1和M2上运行时间的比值是多少?

指令数目相同.

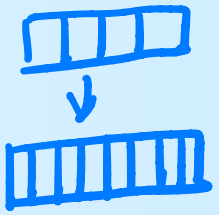
2*（ 1/1.5 ）： 1* （1/1.2）=1.6

CPU执行时间 = 指令数目 x CPI x 一个时钟周期长度

CPU执行时间 — 用于体系结构设计评估



CPU执行时间 = 指令数目 x CPI x 一个时钟周期长度



Improving CPU Performance

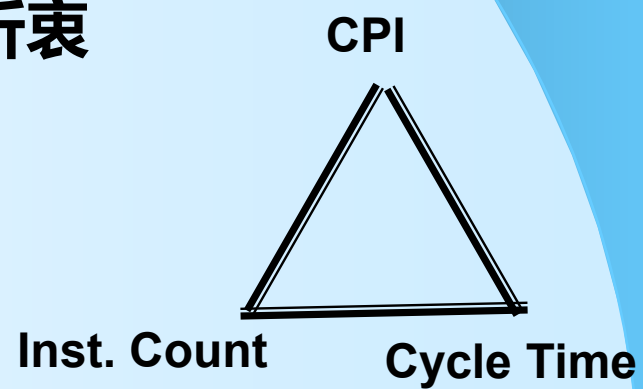
- **IC:**
  - Compiler optimizations (constant folding, constant propagation)
  - ISA (More complex instructions)
- **CPI:**
  - Microarchitecture (Pipelining, Out-of-order execution, branch prediction)
  - Compiler (Instruction scheduling)
  - ISA (Simpler instructions)
- **Clock period:**
  - Technology (Smaller transistors)
  - ISA (Simple instructions that can be easily decoded)
  - Microarchitecture (Simple architecture)

CPU时间三要素

CPU执行时间 = 指令数目 x CPI x 一个时钟周期长度

- 影响CPU时间的三要素
- 计算机系统结构的性能优化方向

  任何一个因子都不能独立地影响性能
- 处理器的性能优化，本质上就是在这三个要素之间折衷



CPU吞吐率 — 性能指标

- **MIPS** —— Million Instruction Per Second
  - 每秒执行百万条指令数
- **FLOPS** —— Floating Point Operation Per Second
  - 每秒浮点运算次数
  - MFLOPS 、GFLOPS、TFLOPS

举例：

❖

测量内容	计算机A	计算机B
指令数	100亿条	80亿条
时钟频率	4GHz	4GHz
CPI	1.0	1.1

哪台计算机MIPS值更高？

$$\frac{1}{1} \times 4 > \frac{1}{1.1} \times 4$$

哪台计算机更快？

$$\frac{1}{4} \times 100 \times 1 > \frac{1}{4} \times 80 \times 1.1$$

MIPS数 = 一周期完成的指令条数 * 时钟频率

CPU执行时间 = 指令数目 * CPI * 一个时钟周期长度

Amdahl 定律

- 加快某部件执行速度所获得的系统性能加速比，受限于该部件在系统中所占的重要性比例。

$$\text{系统总加速比} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}} = \frac{1}{(1-P)+\frac{P}{S}} \Rightarrow \frac{1}{(1-P)}$$

What if I speedup 25% of a program's execution by 2x? 1.14x speedup

What if I speedup 25% of a program's execution by ∞? 1.33x speedup

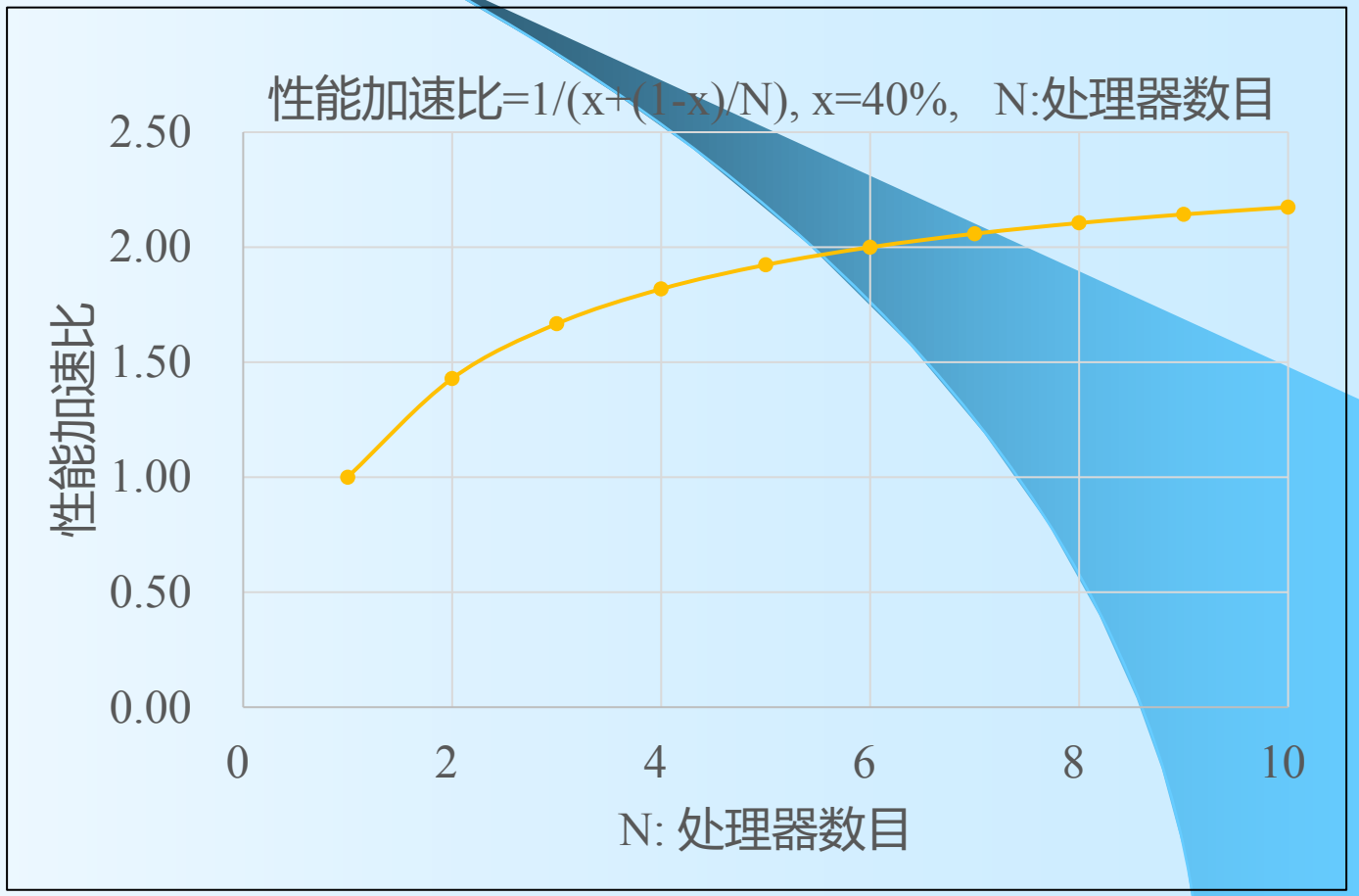
- P = proportion of running time affected by optimization
- S = speedup

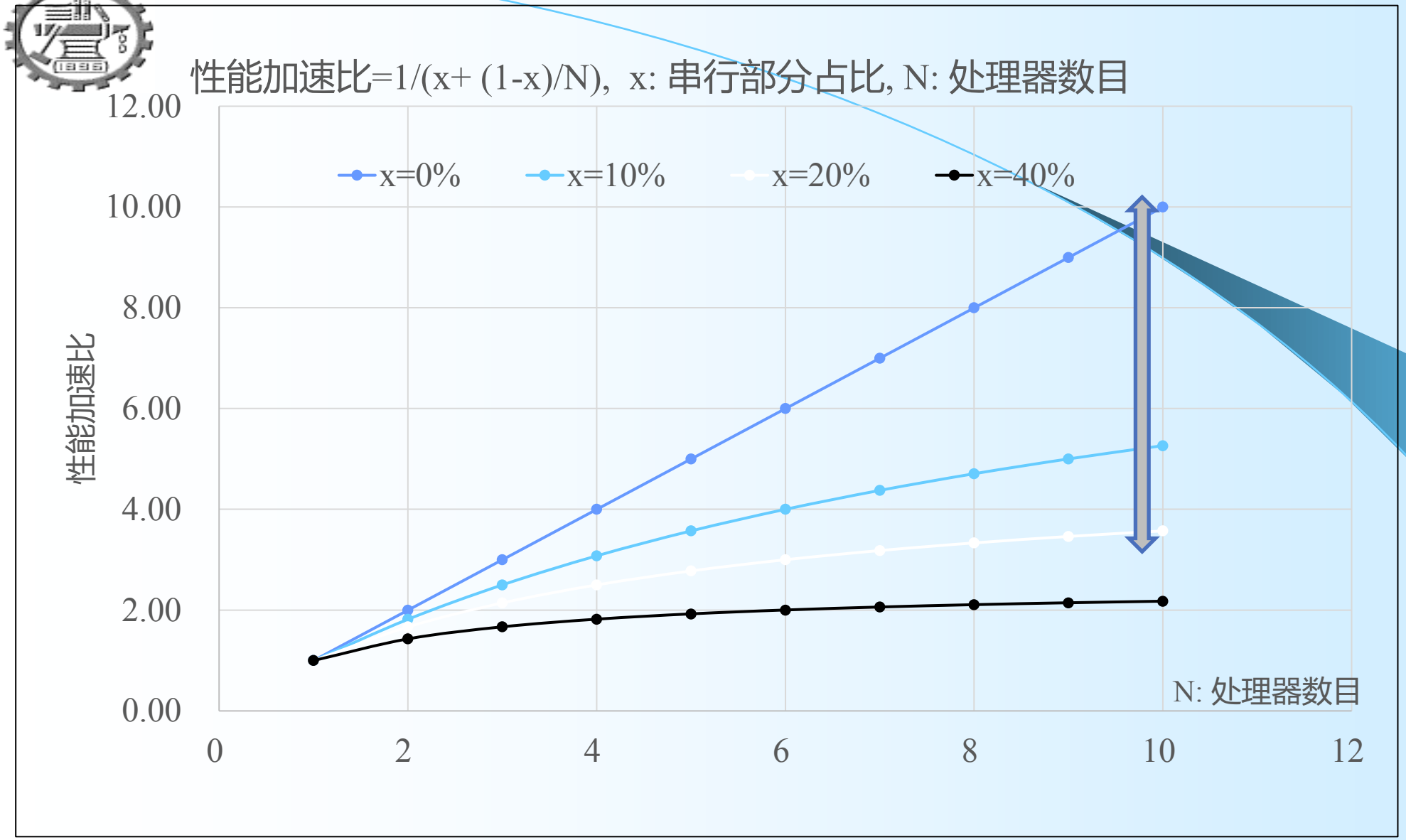
Amdahl 定律

- 如果只针对整个任务的一部分进行优化，那么所获得的加速比是有上限的
- **性能增加的递减规则： 如果仅仅对计算机中的一部分做性能改进，改进越多，系统获得的效果越小**
- **Build a balanced system**
  - **Don’t over-optimize 1% to the detriment of other 99%**
  - **System performance often determined by slowest component**
  - 一个“好”的计算机系统：是一个吞吐率平衡的系统

Amdahl 定律：分析并行计算

- 一个程序的总时间为1；不可并行化部分占40%，可并行化部分就是1 - 0.4 = 0.6；
- 处理器个数：2，加速比：1/(0.4+ 0.6/2) ≈ 1.43
- 处理器个数：5，加速比：1/(0.4+ 0.6/5) ≈ 1.92
- 加速比上限：1/0.4 = 2.5
- 结论：并行计算性能的提升，受到了程序中必须串行执行部分的限制。





理论性能与
实际性能之间
的差距

某些应用的性能确实大幅提升，为什么？

经典的Amdahl 定律并不适用于规模可扩展的并行应用程序的性能分析。

- Amdahl's Law requires *extremely* parallel code to take advantage of large multiprocessors
- two approaches:
  - strong scaling: shrink the serial component
    - same problem runs faster
    - becomes harder and harder to do
  - weak scaling: increase the problem size
    - natural in many problem domains: internet systems, scientific computing, video games
    - doesn't work in other domains

串行: N^2
并行: N^3
当程序扩展, NT. 并行比例↑
不可用 Amdahl 评估.

总结

- 计算机系统结构的研究内容之一：性能评价

- 性能评价指标
  - 运算速度
  - I/O带宽
  - 存储容量
  - 功耗
  - 成本

性能 = 1/执行时间

↓

提升性能

↓

执行时间↓

↓

指令数 × CPI × 1个时钟周期长度