1.4 计算机的性能指标

1.4.3 性能设计的基本原则

主讲人：邓倩妮

上海交通大学

1.4.3 性能设计的基本原则

1. 大概率事件优先的原则

- Make the Common Case Fast： 对于大概率事件（最常见的事件），赋予它优先的处理权和资源使用权，以获得全局的最优结果。

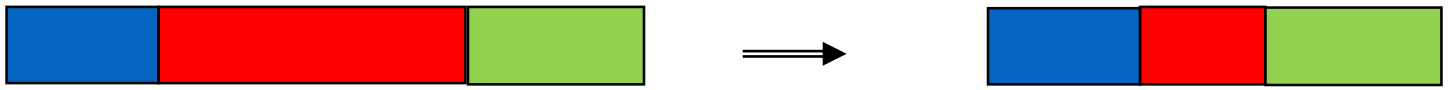
2. Amdahl定律

- 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。

大概率事件优先

(1) 加速比

$$\text{系统加速比} = \frac{\text{系统性能}_{\text{改进后}}}{\text{系统性能}_{\text{改进前}}} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$



Make the Common Case Fast： 对于大概率事件（最常见的事件），赋予它优先的处理权和资源使用权，以获得全局的最优结果。

Amdahl's Law (Amdahl 定律)

$$ExTime_{new} = ExTime_{old} \times \left[(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}} \right]$$

$$Speedup_{overall} = \frac{ExTime_{old}}{ExTime_{new}} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

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，则改进越多，系统获得的效果越小。

Amdahl's Law (Amdahl 定律)

$$\text{Speedup}_{\text{maximum}} = \frac{1}{1 - \text{Fraction time}_{\text{enhanced}}}$$

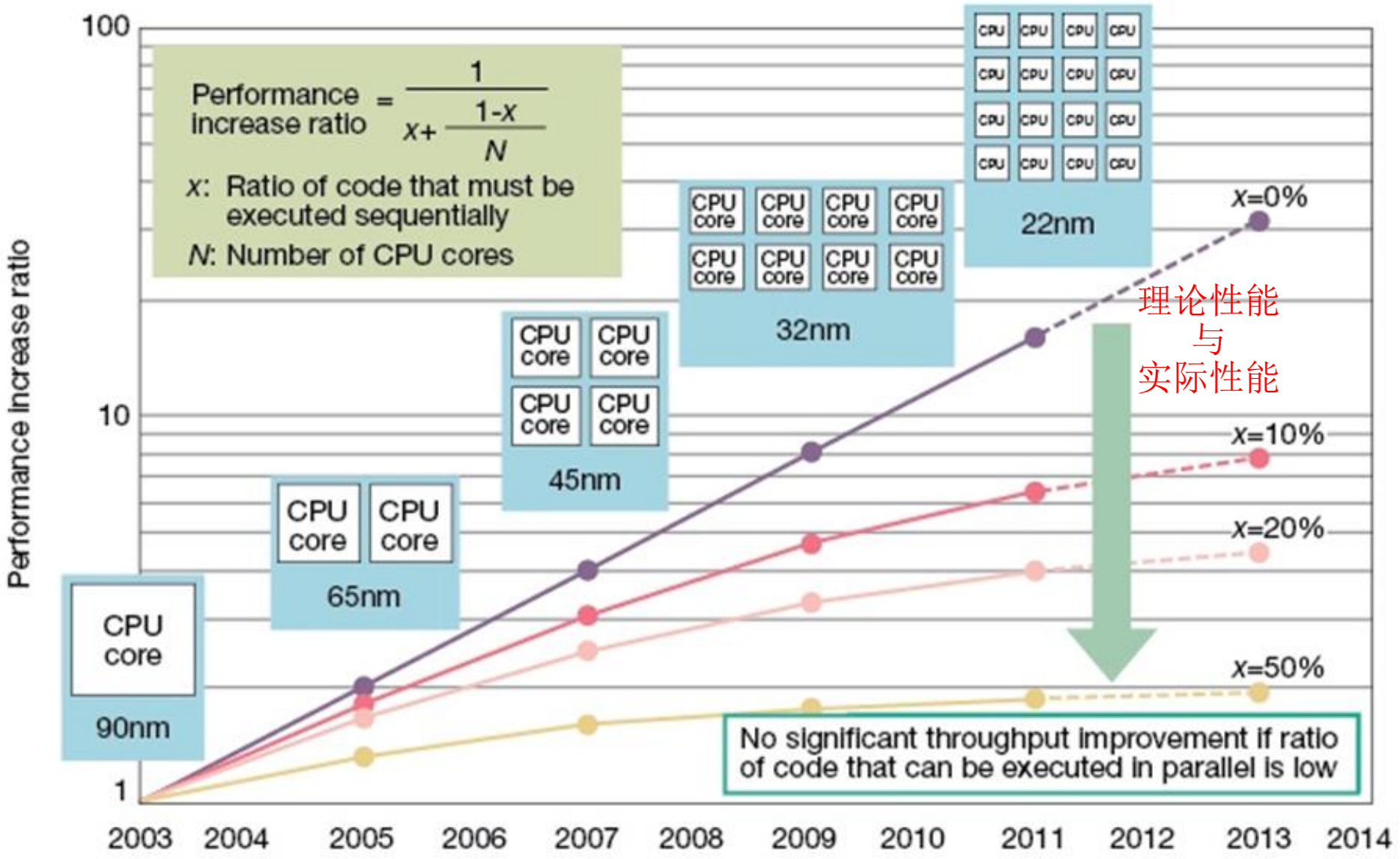
推论：

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不大于1/(1-fe)

练习题：

- 例：我们分析一个用于Web服务器系统的处理器的性能。假定采用某增强方式使新的CPU处理Web服务器应用程序的运行速度是原来处理器中的10倍，同时假定此CPU有40%的时间用于计算，另外60%的时间用于I/O操作。那么增强性能后总的加速比是多少？
  - 解答:
  - Fraction enhanced= 0.4
  - Speedup enhanced=10
  - Speedup overall= 1/(0.6+0.4/10)=1/0.64=1.56

XBOX One


Performance increase ratio = 1 / (x + (1-x)/N)
x: Ratio of code that must be executed sequentially
N: Number of CPU cores
Performance increase ratio
100
10
1
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
CPU core
90nm
65nm
45nm
32nm
22nm
x=0%
x=10%
x=20%
x=50%
理论性能
与
实际性能
No significant throughput improvement if ratio of code that can be executed in parallel is low

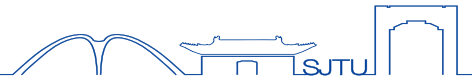
- Amdahl定律揭示了多核处理器中性能的主要障碍:性能受到了软件中必须串行执行部分的影响。当处理器核数目增加，性能没有提升。

- One conclusion made by Amdahl is that "... the effort expended on achieving high parallel processing rates is wasted unless it is accompanied by achievements in sequential processing rates of very nearly the same magnitude."

Gustafson's Law

- In 1988, John Gustafson refined Amdahl's model by adjusting some of its underlying assumptions:
  - There exists workloads that are not in fixed sizes in nature: When provided with more compute power, they expand to consume the newly provided power.
  - When the problem size is increased, the parallel portion expands faster than the serial portion.

Lemma1: Problem size



- Graphics. If I give you more compute power, you will just run your frames at a higher resolution or with more details.
- Numerical analysis such as computing pi. if I give you more compute power, you will just compute more digits of pi.
- Weather Prediction. If I give you more compute power, you will just run your software longer to get even more accurate predictions.

Lemma 2

- When the problem size is increased, the parallel portion expands faster than the serial portion.
  - For example, Martix-Matrix-Multiply (MMM). The setup of MMM, ie.. initializing the matrices increases linearly with the size of the matrix. However, the actual compute is O(n^3).

Gustafson's Law

- Effect of multiple processors on run time of a problem with a fixed amount of parallel work per processor.

$$Speedup(p) = p + (1 - p)s$$

- s is the fraction of non-parallelized code where the parallel work per processor is fixed (not the same as 1-s from Amdahl's)
- p is the number of processors

性能设计原则：小结

- 大概率事件优先的原则： Make the Common Case Fast
- 性能增加的递减规则：只针对整个任务的一部分进行优化，那么所获得的加速比不大于1/(1-fe)
- 一个“好”的计算机系统：具有高性价比的计算机系统是一个带宽平衡的系统，而不是看它使用的某些部件的性能。

谢谢！