计算机组成与设计—绪论2

计算机科学与技术学院

理解计算机性能

理解计算机性能

- 性能="电脑跑得快"
- 主要的量化指标:
  - MIPS (Million Instructions Per Second)
  - FLOPS(Float-point Operations Per Second)
  - CPI (Clock Cycles Per Instruction)
  - IPC (Instructions Per Clock Cycle)
- 实际的性能影响因素
- 关于性能评价

理解计算机性能

- **MIPS：（宏观）每秒执行几百万条指令**

$$MIPS = \frac{被执行的指令条数}{总执行时间(s) \times 10^6}$$

- **MIPS：（微观）核数 x 时钟 x IPC**
  - **IPC, Instructions per Clock，每时钟执行几条指令**

$$MIPS = cores \times clock\ (MHz) \times \frac{Instructions}{clock\ cycle}\ (or\ IPC)$$

理解计算机性能

- **FLOPS，每秒算多少次浮点计算**
  - **在第2章，大家会深刻感受到浮点计算有多麻烦…**
- **Why FLOPS ？**
  - **MIPS更能体现整数计算、应用服务的性能**
  - **FLOPS更能体现科学计算的性能**

$$FLOPS = cores \times clock\ (MHz) \times \frac{FLOPs}{clock\ cycle}\ (or\ FLOPs\ per\ cycle)$$

大小写不同哦！

理解计算机性能

- CPI v.s IPC
  - Cycles Per Instruction (CPI)
  - Instructions Per Cycle (IPC)
- 为什么是两个指标?
  - 早期CPI > 1，后期IPC > 1

理解计算机性能

- IPC、MIPS、FLOPS不是一个确定的值！
  - 不同指令的IPC不一样（指令功能不同，CPU构架不同）
  - 不同指令的执行次数(权重)不同（上层应用功能不同）
- 平均CPI计算
  - 设某CPU提供n条指令，第i条指令的CPI为CPI_i，频率权重为$\frac{IC_i}{IC}$，则该CPU的平均CPI为：

$$\overline{CPI} = \sum_{i=1}^{n} \left(CPI_i \times \frac{IC_i}{IC} \right)$$

理解计算机性能

- **例：CPU主频为20 MHz，设指令集平均CPI为6，求：1. 该CPU的时钟周期，2. 该CPU的MIPS**

$$\text{时钟周期} = \frac{1}{\text{时钟频率}} = \frac{1}{20\ MHz}$$

$$= 0.05 \times 10^{-6}\ S$$

$$\text{平均速度} = \frac{1}{\text{指令执行时间}\quad (S)}$$

$$= \frac{1}{0.05 \times 10^{-6} \times 6} = 3.33\ MIPS$$

计算机性能

- **例：某400 MHz计算机，执行标准测试程序，程序中的指令类型、数量及指令执行的平均周期如下表，求CPI，MIPS及程序执行时间。**

指令类型	指令数量	指令执行周期数
整数	45000	1
数据传送	32000	2
浮点	15000	2
控制传送	8000	2

$$CPI = \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{100000} = \frac{155000}{100000} = 1.55$$

$$MIPS = \frac{Clock\ (MHz)}{CPI} = \frac{400}{1.55} = 258 MIPS$$

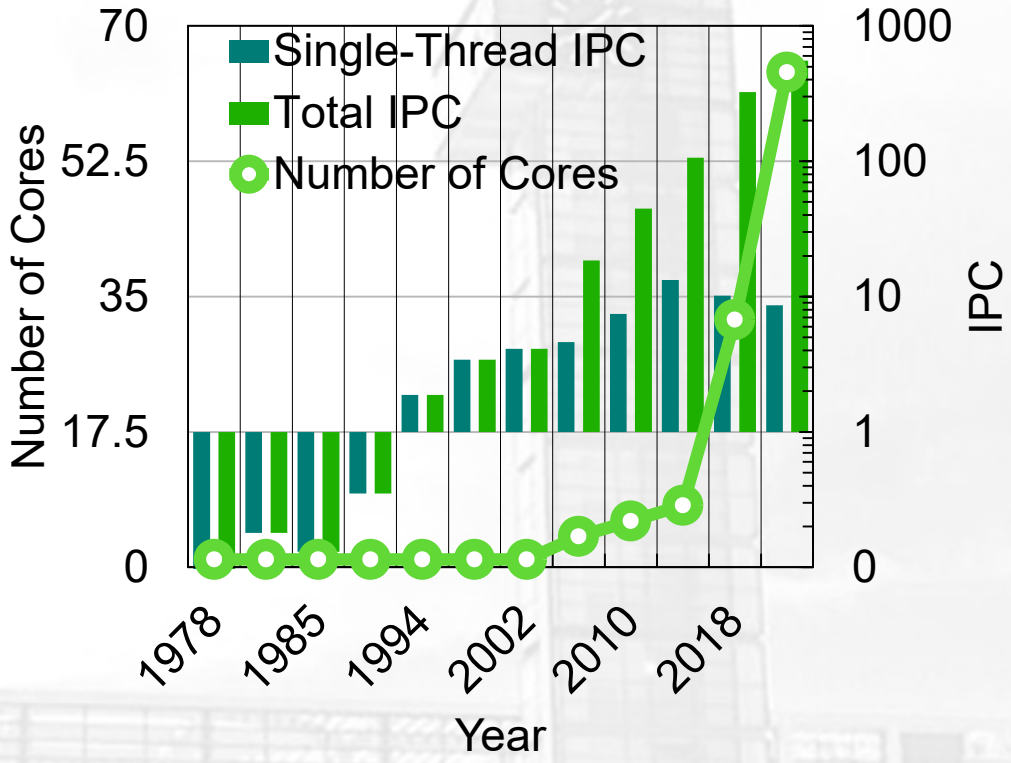
$$执行时间t = \frac{CPI \times 100000}{Clock} = \frac{1.55 \times 100000}{400 \times 1000000} = \frac{1.55}{4000} = 0.00038s = 0.3875ms$$

理解计算机性能

- **MIPS角度来看，现代CPU的单核IPC已大于10！**

CPU MIPS性能提升

年代	型号	主频(MHz)	核心数量	MIPS	总IPC	单核IPC
1978	Intel 8086	5	1	0.33	0.07	0.07
1982	**Motorola 68K**	**12.5**	**1**	**2.188**	**0.18**	**0.18**
1985	Intel i386DX	16	1	2.15	0.13	0.13
1990	Intel i486DX	25	1	8.7	0.35	0.35
1994	Intel Pentium	100	1	188	1.88	1.88
1999	Intel Pentium III	600	1	2054	3.42	3.42
2002	AMD Athlon XP 2500+	1830	1	7527	4.11	4.11
2006	Intel Core 2 QX6700	2660	4	49161	18.48	4.62
2010	Intel Core i7 980X	3300	6	147600	44.73	7.45
2014	**Intel Core i7 5960X**	**3000**	**8**	**317900**	**105.97**	**13.25**
2018	**AMD 2990WX**	**4200**	**32**	**1361770**	**324.23**	**10.13**
2020	**AMD 3990X**	**4340**	**64**	**2394930**	**551.83**	**8.62**



理解计算机性能

- FLOPS角度来看，
  - CPU单周期FLOPs高
    - AVX-256/512, SSE3/4的功劳
  - GPUFLOPs低，所以慢？！
    - Naive，GPU核碾压！
      - AMD 3990x，1.5TFLOPS, 64C/128T
      - Nvidia Titan RTX, 130TFLOPS，4608 + 576(Tensor)

64/32/16bit 浮点计算

CPU FLOPs / Cycle / Core

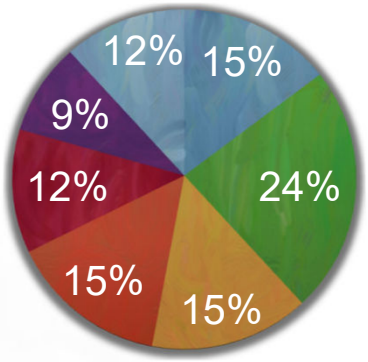
型号	FP64	FP32	FP16
Intel Skylake	16	32	0
Intel Xeon Phi	32	64	0
AMD Zen2	16	32	0
ARM Cortex-A77	8	16	0

GPU FLOPs / Cycle / Core

型号	FP64	FP32	FP16
Nvidia Fermi	1/8	2	0
Nvidia Mexwell	1/16	2	1/32
NVidia Pascal	1	2	4
NVidia Turing	1/16	2	16

理解计算机性能

主要性能影响因素…



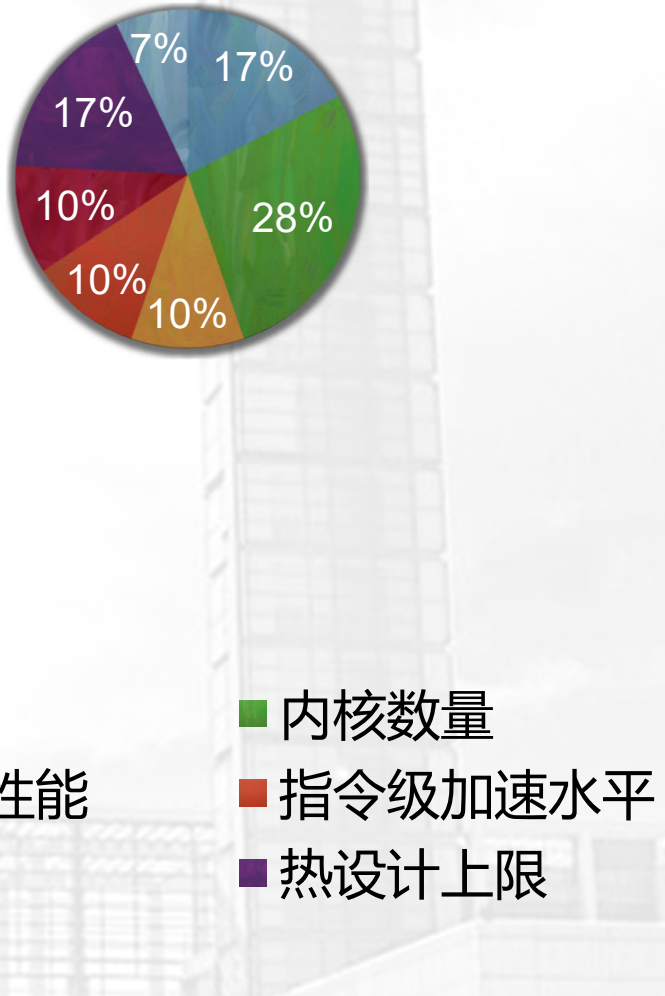
- 主频
- 内核数量
- 单核流水性能
- 指令级加速水平
- I/O及缓存
- 热设计上限
- 其它

- CPU主频对性能的贡献毋庸置疑…
- 核心数量，参考AMD x3990
- 单核流水性能
  - Pentium 4 NetBurst v.s Pentium Core
- 指令级加速水平
  - AVX, SSE, GPU, TPU, FPGA …
- I/O及缓存
  - Ryzen 3 CPU集成传统北桥
  - A13/A12暴锤855/865
- 热设计(Thermal Design Power)
  - 极限TDP的提升
  - 制程提升降低CPU热损
- 其它…

理解计算机性能

- 内核数量，参考x3990
- 主频，参考i3默秒全
- 单核流水性能，参考Core架构持续挤牙膏能力…
- I/O及缓存，大缓存秒天下…
- 指令级加速水平，AVX / NEON
- 热设计，制程提升

可行的性能提升路径…



关于性能评价

- 从性能评测来说，MIPS/FLOPS还是太虚，虚在何处?
  - 影响因素太多:
    - 应用本身的计算倾向
    - 主频
    - CPU类型
    - 操作类型
    - 计算过程
    - 内存访问速度
    - ....

关于性能评价

- 计算机是一个巨系统，有时很难猜测其性能...
  - i3真打不过i7?
  - 64核默秒全?
  - A13打不过i7?
- 有时，连“快”的定义都千差万别：
  - web服务响应高?
  - 吃鸡全程60fps? 😱
  - pi算到1000w位比时间?

那么，怎么比较两个机器性能呢?

那就来跑个分吧!

这是一种病，得治...

迷之自信...😱😱😱666

小米10和笔记本比算圆周率:骁龙865秒杀i7 8550U-小米10..._快科技

2020年2月10日 - 对比机型为:小米10,搭载骁龙865;笔记本,酷睿i7 8550U。测试内容:同采用FFT=ACG算法,计算圆周率小数点后1000万位,统计用时,并核对最后20位数。 结果显...

news.mydrivers.com/1/6... - 百度快照

小米10和笔记本比算圆周率:骁龙865秒杀i7 8550U_凤凰网科技_凤凰网

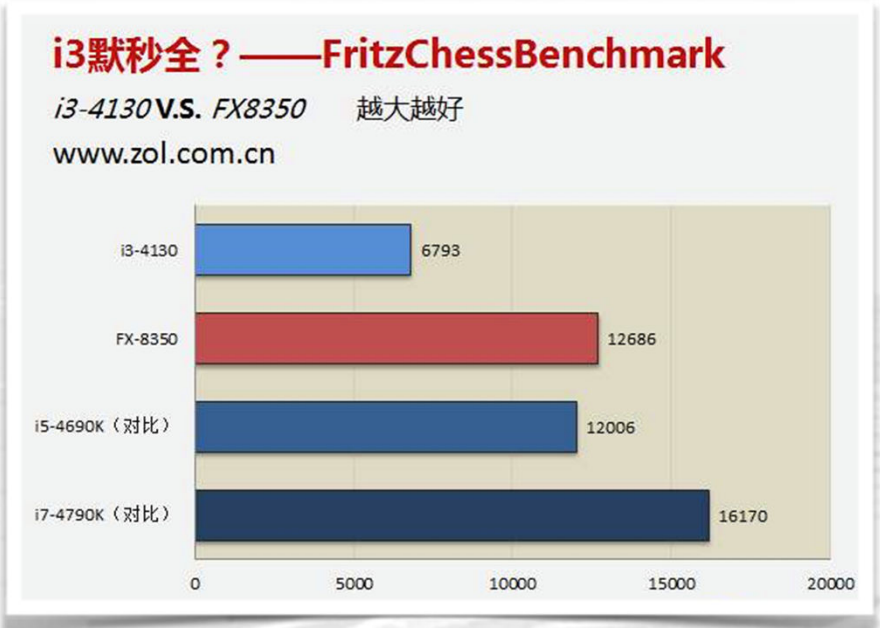
2020年2月10日 - 对比机型为:小米10,搭载骁龙865;笔记本,酷睿i7 8550U。测试内容:同采用FFT=ACG算法,计算圆周率小数点后1000万位,统计用时,并核对最后20位数。 结果显示,小米用时...

手机凤凰网 - 百度快照

小米10和笔记本比算圆周率:骁龙865秒杀i78550U-小米10,笔记本,i7...

今天下午,雷军分享了一段趣味视频:用小米10梦幻性能来计算的一千万位,看看需要多久时间? 对比机型为:小米10,搭载骁龙865;笔记本,酷睿i7 8550U。测试内容:同采用FFT...

www.iwin10.com/itzixun... - 百度快照



苹果A13打得过麒麟990吗?看完秒懂

2019年9月14日 - 在本次发布会上,苹果虽然和友商进行了跑分对比,载A13芯片的iPhone11性能上远远胜过了搭载麒麟980的华为P30 P

TechWeb - 百度快照

麒麟990超越A13 华为报了苹果发布会的"仇"

2019年9月26日 - 发布会现场,华为消费者业务CEO余承东表示:华为990 5G处理器的AI性能,已经超过苹果A13。 这不禁令人们想到了此

中关村在线 - 百度快照

苹果A13芯片集成85亿个晶体管 低于华为麒麟990

2019年9月11日 - A13仿生芯片,是苹果为今秋新推出的iPhone所研发亿个晶体管,较集成69亿个晶体管的A12明显增加,但还是低于华为不

太平洋电脑网 - 百度快照

苹果A13和华为麒麟990到底谁更强?看完你就知道了!

2019年9月12日 - 其实相比A13 Bionic和麒麟980的对比大家更希望这两款先后采用了7nm+EUV工艺的处理器到底谁更胜一筹呢?在9月

科技依依酱 - 百度快照

这是一种病，也得治...

售价2W5：这个电脑玩扫雷肯定不卡

2019-09-15 21:27:45 出处：中关村在线 作者：赵宇航 编辑：万南 人气：8000 次 评论(24)

PC电脑 扫雷

你攒电脑会花多少钱？5千？一万？两万？还是更高？

攒机的时候，需要明确的是如何确认自己的需求。要玩游戏or办公？注重体验还是够用就行？有的人会注重性价比，有的人会注重性能。

如果仅仅是网路游戏的爱好者，大部分情况下不用太高的配置，如果你要加入单机3A戏的行列，那么应该好好考虑一下游戏体验的问题，光2K和1080p分辨率就有差距，更别提低画质和高画质的差距了，顶级配置电脑能一步到位解决大部分用户的需求，而硬件产品花钱就能变强的特点也让我想到了某些企鹅游戏。

接下来介绍一个顶级的配置，这个配置能完美的解决你们你们扫雷会卡的问题。

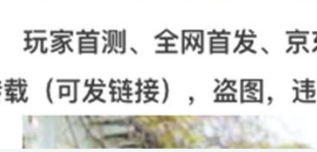
配置清单 参数 兼容与接口

又凭什么不能用扫雷评价性能？ 😱😱😱

GTX980玩扫雷一点都不卡。。。真的。。。

2018-05-22 20:44 来源：网络 作者：网络

玩家首测、全网首发、京东独家，评测文章授权归京东所有，本人一个月后才可以转发转载（可发链接），盗图，违者必究！！



装了一台11万的电脑：扫雷一点不卡！

2015年12月15日 14:18 | 作者：朝晖 | 编辑：孙端 评论 分享

有大侠在吗？请教个问题，我有个朋友，他没工作，失业在家，他爸爸每天只给他一点零花钱。最近他迷上了扫雷，于是想省下一天的零花钱——也就是200万日元吧，准备装台电脑，JS给他推荐了一套平台，但他怕被坑，想问问36核、72线程的双路至强E5+256GB内存+GTX 980 Ti 3-Way SLI的配置够了吗？玩游戏还卡吗？

从此扫雷不再卡 intel 900p 280G体验

2019-04-11 19:49:41 7点赞 8收藏 22评论

三星的SM961用了一两年了做系统盘，妥当当的MLC颗粒，但只有256G，软件装得越来越多，系统的存储也越来越多，D盘是一个500G的2.5寸笔记本硬盘，用惯固态硬盘就回不去了，看到空空如也的PCIE插槽空荡荡，

于是就想搞一个固态硬盘做系统盘，顺便把SM961退休做D盘，JD翻了几页，发现即便过去两年了，感觉三星SM961的王者地位还保持着，不知道这两年M.2 PCIE SSD都在忙啥，感觉一点进步都没有

典型的性能评价非完整列表...

性能评测：比较同一组典型负载的运行时间

- Super Computing
  - HPCG
  - LINPACK

This performance ***does not reflect the overall performance*** of a given system, as no single number ever can. It does, however, ***reflect the performance of a dedicated system for solving a dense system of linear equations***.

- Desktop/Laptop
  - 3D Mark
  - PCMark
  - GeekBench
  - SiSoftware
  - SPEC
  - Cine Bench
  - PassMark
  - SuperPI
  - …

- Mobile
  - GeekBench
  - 3D Mark
  - UNIGINE Benchmarks
  - SiSoftware
  - Antutu 😟 😟 😟

Amdahl's Law
&
Gustafson's Law

Amdahl定律

Amdahl定律内容：计算机系统中某一部件由于采用某种更快的执行方式后，整个系统性能的提高与这种执行方式的使用频率或占总执行时间的比例有关。

人话翻译：把单一部件再怎么优化，也只能提升一点点性能…

G. M. Amdahl：IBM大型机之父

G. M. Amdahl (1922 - 2015)（阿姆达）

《计算机世界》将他列为“改变世界的**25**人”之一。

上世纪六十年代由IBM 360系列计算机的主要设计者。

教育背景：

1948年，获南达科达州立大学工程物理科学学士学位；**1952**年**2**月，获威斯康辛大学理论物理博士学位。

Amdahl定律

计算机系统的加速比S_p取决于下面两个因素：

- 可改进部分在原系统总执行时间中的占比，称为可改进比例，用f_e表示。
  - 例如，程序的总执行时间为100s，可改进的部分是其中的20s，则$f_e = 0.2$。 fe总是小于或等于1。
- 可改进部分改进后性能提高的程度，用r_e表示。
  - 例如，某部件执行时间由原来的20s减少到5s，则部件加速比$r_e = 20/5 = 4$。re一般大于1。

Amdahl定律

- 若假设改进前的系统总执行时间为T_o，可以得出改进后的系统总执行时间T_n为：

$$T_n = T_o\left(1 - f_e + \frac{f_e}{r_e}\right)$$

- 若加速比用S_p表示，则加速比S_p可表示为：

$$S_p = \frac{1}{(1 - f_e) + \frac{f_e}{r_e}}$$

- 当系统可改进的部分f_e确定后，既使这一部分改进后不再需要时间，即$r_e \to \infty$，则Sp = 1/(1 - f_e)。

Amdahl定律

【例1.1】某计算机系统的某一部件的处理时间为总处理时间的40%，该部件改进后部件加速比为10，试计算改进后系统的加速比S_p为多少。

解：由上述题意可知，$f_e=0.4$，$r_e=10$，则

$$S_p=1/[(1-0.4)+0.4/10]\approx 1.56$$

由计算可见，既使某一部件的部件加速比已达10倍，但该部件仅影响到总执行时间的小部分，对整个计算机系统的贡献是有限的。所以，改进后系统的加速比只有1.5倍左右。

Amdahl定律

【例1.2】若计算机系统有三个部件a、b、c是可改进的，它们的部件加速比分别为30、30、20。它们在总执行时间中所占的比例分别是30%、30%、20%。试计算这三部件同时改进后系统的加速比。

Amdahl定律

【例1.2】若计算机系统有三个部件a、b、c是可改进的，它们的部件加速比分别为30、30、20。它们在总执行时间中所占的比例分别是30%、30%、20%。试计算这三部件同时改进后系统的加速比。

解：多个部件可同时改进的情况下，Amdahl定律可表示为：

$$S_p = \frac{1}{(1-\sum f_e)+\sum\left(\frac{f_e}{r_e}\right)}$$

$$S_p = \frac{1}{(1-0.3-0.3-0.2)+\left(\frac{0.3}{30}+\frac{0.3}{30}+\frac{0.2}{20}\right)} \approx 4.35$$

Gustafson定律

Gustafson定律内容：当计算任务可足够并行时，加速比随着计算部件的数目而线性增长。

人话翻译：内核越多越爽…

Gustafson定律

Dr. John L. Gustafson（古斯塔夫森）

爱荷华大学教授

前AMD首席产品构架师

曾在Intel, Sun任职

教育背景：

1977年，获加州理工学院应用数学学士学位；**1982**年获爱荷华州立大学博士学位。

Gustafson定律

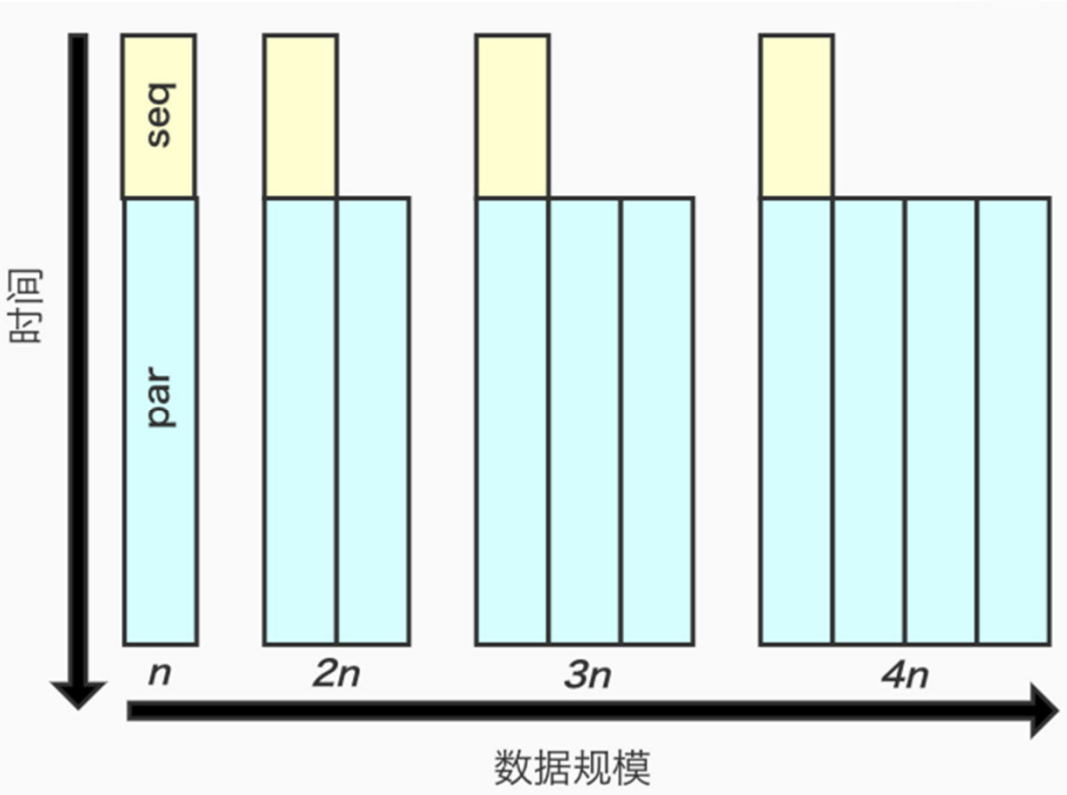
Gustafson定律：

总任务中，必须顺序执行的部分为f, 在单处理器上顺序执行的时间为t_s；设有p个并行处理器参与计算，则通过并行化处理，就可以获得接近p的加速比：

$$S(p) = \frac{f t_s + (1-f)t_s}{f t_s + (1-f)t_s/p}$$

Gustafson定律

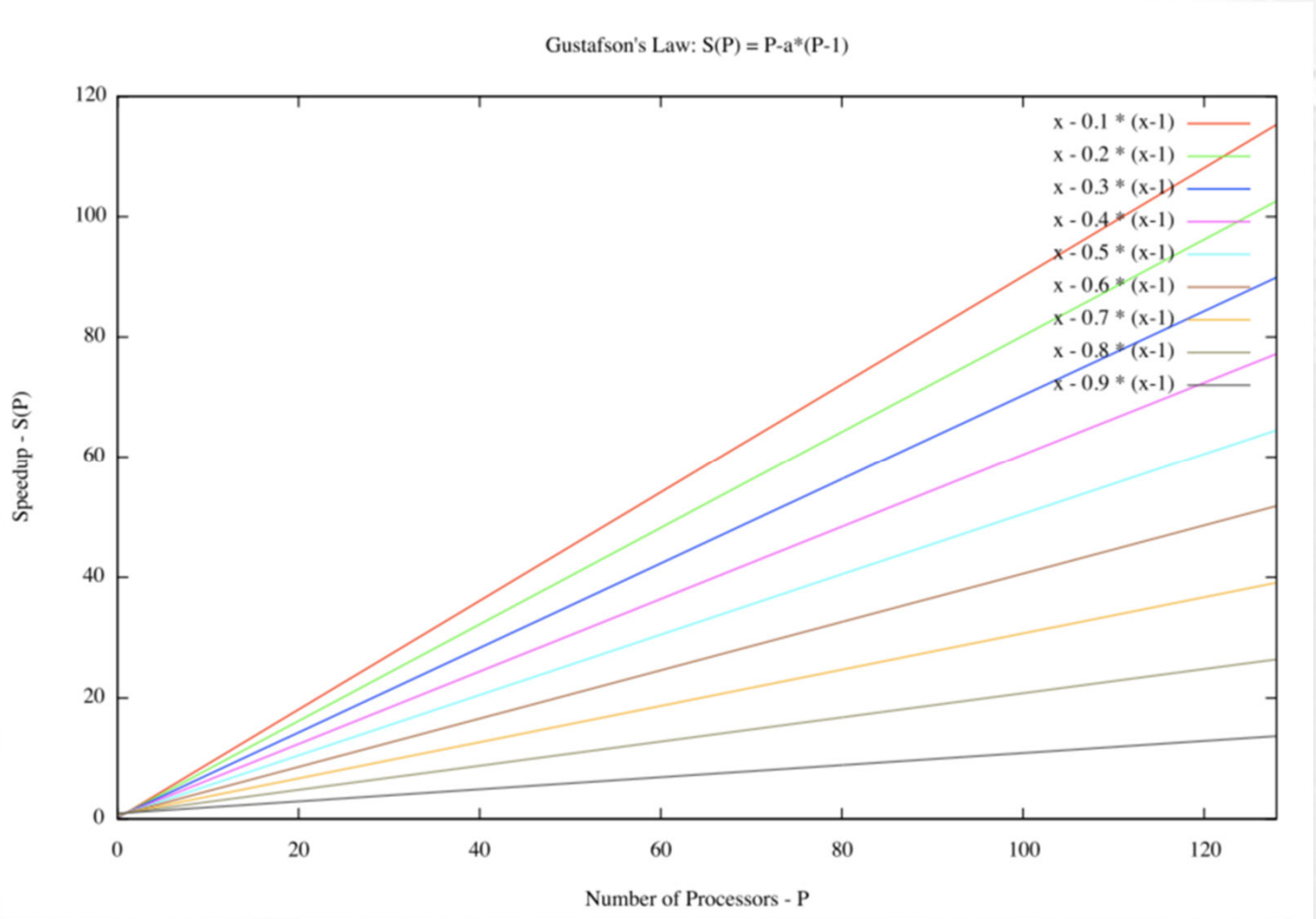
总任务中，必须顺序执行的部分为f, 在单处理器上顺序执行的时间为t_s；设有p个并行处理器参与计算，可以用相同时间处理的数据量来衡量加速比



$$S(p) = f + (1 - f)p$$

串行部分代码比例固定的前提加，加速比会随着处理器个数增加而增加。(线性增加)

Gustafson定律



$$S(p) = f + (1 - f)p$$

串行部分代码比例固定的前提加，加速比会随着处理器个数增加而增加。(线性增加)

哪个更有启发意义?

- Gustafson's Law为计算机性能提升给出了关键指导:
  - 处理器要多，很多，非常多，非常非常多…
  - I/O要快，很快，非常快，非常非常快…
  - 任务要最大化分解
    - 能并行不串行…
    - 能低效并行不高效串行…
    - 小到GPU渲染…
    - 大到超算…

举个例子

- 实现int类型矩阵加法，该过程可分解以及消耗指令数为：
- 读取数据，需要10条指令
- 计算结果，其中每次int加法需要4条指令
- 存储数据，需要20条指令
- 计算机的CPU性能为10MIPS，如果为计算机加装上限1000核心的GPU，其每个核心的性能为0.1MIPS，
- 计算采用GPU加速后10*10大小矩阵加法耗时，以及加速比
- 计算采用GPU加速后100*100大小矩阵加法耗时，以及加速比
- 对于10*10大小矩阵：
- CPU运算耗时为：(10+20+4*100)/10=43us
- CPU+GPU: (10+20)/10+4/0.1=43us
- 加速比：43/43=1
- 对于100*100大小矩阵：
- CPU运算耗时为：(10+20+4*10000)/10=4003us
- CPU+GPU: (10+20)/10+4*(10000/1000)/0.1=403us
- 加速比：4003/403=9.93

cuda

```
1  __global__
2  void vecAddKernel(float* A_d, float* B_d, float* C_d, int n)
3  {
4      // GPU中执行并行计算的thread id
5      // 类比CPU中计算数组位置
6      int i = threadIdx.x + blockDim.x * blockIdx.x;
7      if (i < n) C_d[i] = A_d[i] + B_d[i];
8  }
```

矩阵加法

GPU并行核心

第一章课后习题

- 10
- 13
- 3月06号之前在西电课堂提交