



中国科学技术大学
University of Science and Technology of China

COD HW1

蔡豪语

中国科学技术大学计算机学院
嵌入式系统实验室
2020年春

内容提要



中国科学技术大学
University of Science and Technology of China

□1.14

□1.16

习题 1. 14

1.8 节引证了一个用性能公式的一个子集去计算性能的陷阱。为了说明它，下表是在不同的处理器中执行 10^6 条指令序列的有关数据。

处理器	时钟频率	CPI
P1	4 GHz	1.25
P2	3 GHz	0.75

1.14.1 [5] <1.8> 一个常见的错误是，认为时钟频率最高的计算机具有最高的性能。这种说法正确吗？请用 P1 和 P2 来验证这一说法。

不正确 当P1和P2均执行 10^6 条指令时 **假设P1与P2的指令系统相同**

$$T_{CPU} = \frac{IC_i \times CPI_i}{f}, \text{ 则有 } T_{CPU_1} = 3.125 \times 10^{-4} s, T_{CPU_2} = 2.5 \times 10^{-4} s, T_{CPU_1} > T_{CPU_2}, \text{ 即}$$

P2 虽然时钟频率低，但拥有更高的性能

1.14.2 [10] <1.8> 另一个错误是，认为执行指令最多的处理器需要更多的 CPU 时间。考虑 P1 执行 10^6 条指令序列所需的时间，P1 和 P2 的 CPI 不变，计算一下 P2 用同样的时间可以执行多少条指令？

$$n_2 = \frac{n_1 T_1}{T_2} = \frac{10^6 \times 3.125}{2.5} = 1.25 \times 10^6, \text{ P2 用同样的时间可以执行 } 1.25 \times 10^6 \text{ 条指令}$$

运算速度



中国科学技术大学
University of Science and Technology of China

- 衡量计算机运算速度的指标，与许多因素有关，如机器主频、操作类型、主存速度等。
- 基准程序法 (benchmark)
 - ✓ 考虑CPU、I/O结构、操作系统、编译器效率等
 - ✓ 评价计算机的实际工作能力
- 目前常用的测试程序：
 - ✓ 真实程序：C编译器、Matlab、Photoshop… (移植带来问题)
 - ✓ **核心程序**：由从真实程序中提取的较短但很关键的代码构成。
Livermore loops、Linpack
 - ✓ 合成测试程序 (Whetstone/Dhrystone)
 - ✓ **测试程序组** (**测试程序包**：Benchmarks)：选择一组各个方面有代表性的测试程序，组成的一个通用测试程序集合。



嵌入式系统实验室
EMBEDDED SYSTEM LABORATORY
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

习题 1. 14

1.8 节引证了一个用性能公式的一个子集去计算性能的陷阱。为了说明它，下表是在不同的处理器中执行 10^6 条指令序列的有关数据。

处理器	时钟频率	CPI
P1	4 GHz	1.25
P2	3 GHz	0.75

1. 14. 3 [10] <1.8> 一个常见的错误是用 MIPS（每秒百万条指令）来比较 2 台不同的处理器的性能，并认为 MIPS 最大的处理器具有最高的性能。这种说法正确吗？请用 P1 和 P2 验证这一说法。

$$MIPS = \frac{n_{MIPS}}{T_{CPU} \times 10^6}$$

则对处理器 P1, $MIPS_1 = 3.2 \times 10^3 MIPS$

对程序 b, $MIPS_2 = 4 \times 10^3 MIPS$

如果两者没有相同的指令系统，只依据MIPS判断性能高低是没有意义的。

例如，p1和p2执行功能相同的程序，由于p1的指令集结构和编译技术更科学，在处理器p1中该程序的功能只需要1000条指令就能完成；而在p2中需要10000条指令才能完成相同的功能。

这种情况下p1显然有更高的性能，即使p2有着MIPS指标更高。

运算速度



中国科学技术大学
University of Science and Technology of China

存在问题

编辑

MIPS是指令执行的速率，规定了性能和执行时间成反比，越快的计算机具有越高的MIPS值。从表面看，MIPS既容易理解，又符合人的直觉。其实，用MIPS作为度量性能的指标存在三个问题。

首先，MIPS规定了指令执行的速率，但没有考虑指令的能力。我们没有办法用MIPS比较不同指令集的计算机，因为指令数肯定是不一样的。

其次，在同一计算机上，不同的程序会有不同的MIPS，因而一台计算机不会只有一个MIPS值。例如，将执行时间用MIPS、CPI、时钟频率代入之后可得：

$$\text{MIPS} = \text{指令数} / (\text{执行时间} * 10^6) = \text{指令数} / (\text{指令数} * \text{CPI} / \text{时钟频率} * 10^6) = \text{时钟频率} / (\text{CPI} * 10^6)$$

Description	Name	Instructions Executed x 10 ⁶	CPI	Clock cycle time (seconds x 10 ⁻⁹)	Execution Time (seconds)	Reference Time (seconds)	SPS/ratio
integrated string processing	perl	2252	0.80	0.876	588	4770	14.2
block-matching	nmcli	2280	0.73	0.876	629	4650	13.4
compression							
gzip	gzip	764	1.20	0.876	386	4650	22.5
Combinatorial optimization	net	221	2.68	0.876	221	4287	45.2
Go game (k)	go	1274	5.10	0.876	527	4340	14.9
Search game sequence	hmmcr	2630	0.60	0.876	590	4330	15.6
Chess game (k)	chess	1444	1.90	0.876	586	4230	22.7
Quasimc computer simulation	Quasimc	480	0.83	0.876	136	2070	280.2
Video compression	h264enc	1070	0.50	0.876	713	2230	25.0
Divide-and-conquer	amrmp	367	2.10	0.876	280	4280	21.0
Quasimc computer simulation	Quasimc	1250	1.00	0.876	407	7020	14.9
Path finding	edmonds	1040	0.70	0.876	276	4800	26.1
Quasimc computer simulation	Quasimc	-	-	-	-	-	26.7

上图显示了SPEC2006在Intel Core i7上的CPI最大值和最小值是相差5倍的，MIPS也是如此。

下表是linux native的benchmark，除了x86版本外，也都能找到或者编译出arm版本。

工具名称	测试项目	#####summary#####	项目地址
coremark	cpu	测评cpu的整体性能（列举、矩阵运算、状态机、CRC）	
coremark_pro	cpu	coremark的升级版，测评cpu整体性能	
super PI	cpu	测评PI的计算	ftp://pi.super-computing.org/Linux/
SPEC	cpu	测评cpu性能	http://www.spec.org/spec/
dhystone	cpu	测评CPU整形计算	
whetstone	cpu	测评CPU浮点运算	
nbench	cpu&mem	测评CPU运算性能（整数运算、双精度浮点运算）/mem指数主要体现处理器总线、cache和存储器性能	http://www.tux.org/~mayer/linux/bmark.html
utest_mem	mem	测评mem bandwidth	
cachebench	mem	测评mem&cache bandwidth	http://icl.cs.utk.edu/projects/lcbench/cachebench.html
copybw	mem	测评mem bandwidth	http://www.tux.org/pub/benchmarks/CPU/copybw.c
ramspeed	mem	测评cache有效带宽	
bonnie	IO	测评IO性能	http://www.textuality.com/bonnie/
Fio	IO	测评IO性能	http://freshmeat.net/projects/fio/
Imbench	CPU/mem/IO	测评cpu/mem/IO bandwidth & latency	
sysbench	CPU/mem/IO	多线程性能测试	

SPEC是标准性能评估公司（Standard Performance Evaluation Corporation）的简称。SPEC是由计算机厂商、系统集成商、大学、研究机构、咨询等多家公司组成的非营利性组织，这个组织的目标是建立、维护一套用于评估计算机系统的标准。

在早些年，业界使用的是其上一个版本SPEC CPU 2000。和SPEC CPU 2000一样，SPEC CPU 2006包括了CINT2006和CFP2006两个子项目，前者用于测量和对比整数性能，而后者则用于测量和对比浮点性能，SPEC CPU 2006中对SPEC CPU 2000中的一些测试进行了升级，并抛弃/加入了一些测试，因此两个版本测试得分并没有可比较性。

SPEC CPU测试中，测试系统的处理器、内存子系统和使用到的编译器（SPEC CPU提供的是源代码，并且允许测试用户进行一定的编译优化）都会影响最终的测试性能，而I/O（磁盘）、网络、操作系统和图形子系统对于SPEC CPU2006的影响非常的小。

An ounce of honest data is worth a pound of marketing hype（一盎司诚实的数据值得一磅的市场宣传）是SPEC组织成立的座右铭，为了保持数据的公平、可信度以及有效，SPEC CPU测试使用了现实世界的应用程序，而不是用循环的算术操作来进行基准测试。SPEC CPU 2006包括了12项整数运算和17项浮点运算，除此之外，还有两个随机数产生测试程序998.sperand（整数）和999.specrand（浮点），它们虽然也包含在套件中并得到运行，但是它们并不进行计时以获得得分。这两个测试主要是用来验证一些其他组件中会用到的PRNG随机数生成功能的正确性。各个测试组件基本上由C和Fortran语言编写，有7个测试项目使用了C++语言，而Fortran语言均用来编写浮点部分。

CINT2006包括C编译程序、量子计算机仿真、下象棋程序等，CFP2006包括有限元模型结构化网格法、分子动力学质点法、流体动力学稀疏线性代数法等。为了简化测试结果，SPEC决定使用单一的数字来归纳所有12种整数基准程序。具体方法是将被测计算机的执行时间标准化，即将被测计算机的执行时间除一个参考处理器的执行时间，结果称为SPECratio。SPECratio值越大，表示性能越快（因为SPECratio是执行时间的倒数）。CINT2006或CFP2006的综合测试结果是取SPECratio的几何平均值。

3.3 运算速度



中国科学技术大学
University of Science and Technology of China

□ 早期衡量运算速度的普通方法：完成一次加法或乘法所需的时间

✓ CPI (Cycle per Instruction)

- 指令周期，表示执行一条指令所需的平均时钟周期数

$$CPI = \frac{\text{执行程序所需的CPU时钟周期数}}{\text{程序包含的指令条数}}$$

✓ MIPS (Million Instruction Per Second)

- 百万条指令每秒，即单位时间内执行的指令数

$$MIPS = \frac{\text{程序的指令条数}}{\text{程序的执行时间} \times 10^6}$$

✓ MFLOPS (Million Floating Point Operation Per Second)

- 百万次浮点操作每秒，用来衡量机器浮点运算的性能

$$MFLOPS = \frac{\text{程序的浮点操作次数}}{\text{程序的执行时间} \times 10^6}$$



嵌入式系统实验室
EMBEDDED SYSTEM LABORATORY
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

1.14



另一个常见的性能标志是 MFLOPS（每秒百万条浮点指令），其定义为 $\text{MFLOPS} = \text{浮点操作的个数} / \text{执行时间} \times 10^6$ 。它与 MIPS 有同样的问题。考虑下表所示的程序，在时钟频率为 3 GHz 的处理器上运行。

	指令数	读/写	浮点	分支	CPI（读/写）	CPI（浮点）	CPI（分支）
程序 a	10^6	50%	40%	10%	0.75	1	1.5
程序 b	3×10^6	40%	40%	20%	1.25	0.70	1.25

1.14.4 [10] <1.8> 求程序的 MFLOPS 值。

1.14.5 [10] <1.8> 求程序的 MIPS 值。

1.14.6 [10] <1.8> 求程序的性能，并与 MFLOPS 和 MIPS 值作比较。

1.14.4

$$T_{CPU} = \sum \frac{IC_i \times CPI_i}{f}$$

则对程序 a, $T_{CPUa} \approx 3.083 \times 10^{-4} s$, 对程序 b, $T_{CPUb} = 1.03 \times 10^{-3} s$

$$\text{又 } MFLOPS = \frac{n_{MFLOPS}}{T_{CPU} \times 10^6}$$

则对程序 a, $MFLOPS_a = 1.2973 \times 10^3 MFLOPS$

对程序 b, $MFLOPS_b = 1.165 \times 10^3 MFLOPS$

1.14



另一个常见的性能标志是 MFLOPS（每秒百万条浮点指令），其定义为 $\text{MFLOPS} = \text{浮点操作的个数} / \text{执行时间} \times 10^6$ 。它与 MIPS 有同样的问题。考虑下表所示的程序，在时钟频率为 3 GHz 的处理器上运行。

	指令数	读/写	浮点	分支	CPI（读/写）	CPI（浮点）	CPI（分支）
程序 a	10^6	50%	40%	10%	0.75	1	1.5
程序 b	3×10^6	40%	40%	20%	1.25	0.70	1.25

1.14.4 [10] <1.8> 求程序的 MFLOPS 值。

1.14.5 [10] <1.8> 求程序的 MIPS 值。

1.14.6 [10] <1.8> 求程序的性能，并与 MFLOPS 和 MIPS 值作比较。

1.14.5

$$\text{MIPS} = \frac{n_{\text{MIPS}}}{T_{\text{CPU}} \times 10^6}$$

则对程序 a, $\text{MIPS}_a = 3.24 \times 10^3 \text{ MIPS}$

对程序 b, $\text{MIPS}_b = 2.9126 \times 10^3 \text{ MIPS}$

1.14.6

由 1.14.4 知对程序 a, $T_{\text{CPU}a} \approx 3.083 \times 10^{-4} \text{ s}$, 对程序 b, $T_{\text{CPU}b} = 1.03 \times 10^{-3} \text{ s}$

程序 a 的性能要优于程序 b, 使用 MIPS 与 MFLOPS 得出的结果同样是程序 a 更好

$\text{Performance} = 1/T$ $\text{Performance}_a = 3.243 \times 10^3$; $\text{Performance}_b = 0.971 \times 10^3$



1.16



还有一个易犯的错误是有关在多处理器系统中运行，希望只改进一部分例行程序来改进整体性能。下表表示某个程序的 5 个例程在不同数目处理器中的执行时间。

	处理器数	例程 A (ms)	例程 B (ms)	例程 C (ms)	例程 D (ms)	例程 E (ms)
a.	2	20	80	10	70	5
b.	16	4	14	2	12	2

1.16.1 [10] <1.8> 求总的执行时间。如果例程 A, C 和 E 的时间改进 15%，总的执行时间能减少多少？

1.16.2 [10] <1.8> 如果例程 B 的时间改进 10%，总的执行时间能减少多少？

1.16.3 [10] <1.8> 如果例程 D 的时间改进 10%，总的执行时间能减少多少？

1.16.1

处理器数为 2 时, $T_a = 185ms$, 处理器数为 16 时 $T_b = 34ms$

处理器数为 2 时, $\Delta T_{a1} = 0.15 \times (20 + 10 + 5)ms = 5.25ms$

处理器数为 16 时, $\Delta T_{b1} = 0.15 \times (4 + 2 + 2)ms = 1.2ms$

1.16.2

处理器数为 2 时, $\Delta T_{a2} = 0.1 \times 80ms = 8ms$

处理器数为 16 时, $\Delta T_{b2} = 0.1 \times 14ms = 1.4ms$

1.16.3

处理器数为 2 时, $\Delta T_{a3} = 0.1 \times 70ms = 7ms$

处理器数为 16 时, $\Delta T_{b3} = 0.1 \times 12ms = 1.2ms$

1.16



多处理器系统中的执行时间可分成例程计算时间加处理器之间的通信时间。下表给出了例程计算时间和通信时间。在这种情况下，通信时间是总时间的重要组成部分。

处理器数	例程 A (ms)	例程 B (ms)	例程 C (ms)	例程 D (ms)	例程 E (ms)	通信 (ms)
2	20	78	9	65	4	11
4	12	44	4	34	2	13
8	1	23	3	19	3	17
16	4	13	1	10	2	22
32	2	5	1	5	1	23
64	1	3	0.5	1	1	26

1.16.4 [10] <1.8> 每当处理器数量加倍时，求新的计算时间与旧的计算时间之比和新与旧的通信时间之比。

1.16.5 [5] <1.8> 用比值的几何平均值，推算在 128 台处理器的系统中的计算时间和通信时间。

1.16.6 [10] <1.8> 求在 1 个处理器系统中的计算时间和通信时间。

处理器数	计算时间	通信时间	计算时间比	通信时间比
2	176	11	-	-
4	96	13	0.5455	1.1818
8	49	17	0.5104	1.3077
16	30	22	0.6122	1.2941
32	14	23	0.4667	1.0455
64	6.5	26	0.4643	1.1304
128	3.36	30.88	0.517	1.1877
1	340.43	0	-	-

