

湖南大学

HUNAN UNIVERSITY

计算机体系结构 作业

项目名称： 习题

学生姓名： 王鑫淼

学生学号： 201508010513

专业班级： 智能 1501 班

完成日期： 2018 年 12 月 26 日

1、

- 1.5 [10/10/20] <1.5>在为服务器场供电时，一个至关重要的因素就是冷却。如果不能有效地使计算机散热，风扇就会把热空气而不是冷空气吹回计算机。我们将研究各种不同设计决策如何影响一个系统的必要冷却方式，从而影响它的价格。请使用表 1-12 进行功率计算。
- a. [10] <1.5>机架上安装一个冷却门的成本是 4000 美元，可以消散 14 KW（向室内散热；向室外散热需要增加成本）。如果服务器采用 Intel Pentium 4 处理器，1 GB 240 管脚 DRAM 和一个 7200 rpm 硬盘驱动器，那么一个冷却门可以冷却多少个服务器？
- b. [10] <1.5>我们正在考虑为硬盘驱动器提供容错功能。RAID 1 将磁盘数目加倍（见第 6 章）。现在，一个只有一个冷却器的机架中可以放入多少个系统？
- c. [20] <1.5>典型的服务器场每平方英尺可以消耗最多 200W 的热量。如果一个服务器机架需要 11 平方英尺（包括前后间隙），在一个机架内可以放入多少个第(a)步中提到的服务器？需要多少个冷却门？

表1-12 几种计算机组件的功耗

组件类型	产 品	性 能	功 率
处理器	Sun Niagara 8-core	1.2 GHz	72~79 W（峰值）
	Intel Pentium 4	2 GHz	48.9~66 W
DRAM	金士顿X64C3AD2 1 GB	184个管脚	3.7W
	金士顿D2N3 1 GB	240个管脚	2.3W
硬盘驱动器	DiamondMax 16	5400 rpm	读取/寻道时7.0 W，空闲时2.9 W
	DiamondMax 9	7200 rpm	读取/寻道时7.9 W，空闲时4.0 W

解：

- a) 服务器数量： $n = \frac{14\text{kW}}{66\text{W} + 2.3\text{W} + 7.9\text{W}} = 183.7 \approx 183$ 个
- b) 系统个数： $m = \frac{14\text{kW}}{66\text{W} + 2.3\text{W} + 2 * 7.9\text{W}} = 166.5 \approx 166$ 个
- c) 每个服务器消耗： $200\text{W} * 11 = 2200\text{W}$
- 一个机架内： $\frac{2200\text{W}}{66\text{W} + 2.3\text{W} + 7.9\text{W}} = 28.87 \approx 28$ 个

由于 2.2kW 小于 14kW，所以只需要一个冷却门。

2、

- 1.18 [10/20/20/20/25] <1.10>在实现一个应用程序的并行化时，理想加速比应当等于处理器的个数。但它要受到两个因素的限制：可并行化应用程序的百分比和通信成本。Amdahl 定律考虑了前者，但没有考虑后者。
- a. [10] <1.10>如果应用程序的 80%可以并行化， N 个处理器的加速比为多少？（忽略通信成本。）
- b. [20] <1.10>如果每增加一个处理器，通信开销为原执行时间的 0.5%，则 8 个处理器的加速比为多少？
- c. [20] <1.10>如果处理器数目每增加一倍，通信开销增加原执行时间的 0.5%，则 8 个处理器的加速比为多少？
- d. [20] <1.10>如果处理器数目每增加一倍，通信开销增加原执行时间的 0.5%，则 N 个处理器的加速比为多少？
- e. [25] <1.10>写出求解这一问题的一般公式：如果一个应用程序中占原执行时间的 $P\%$ 可以并行化，处理器数目每增加一倍，通信成本增加原执行时间的 5%，则达到最高加速比的处理器数目为多少？

解：

$$a) \text{加速比为: } \frac{1}{0.2 + \frac{0.8}{N}} = \frac{N}{0.2N + 0.8}$$

$$b) \text{加速比为: } \frac{1}{0.2 + 8 * 0.005 + \frac{0.8}{8}} = 2.941$$

$$c) 8 \text{ 个处理器的加速比: } \frac{1}{0.2 + \log_2 8 * 0.005 + \frac{0.8}{8}} = 3.17$$

$$d)) 8 \text{ 个处理器的加速比: } \frac{1}{0.2 + \log_2 N * 0.005 + \frac{0.8}{N}}$$

$$e) \text{加速比: } \frac{1}{1 - P\% + \log_2 N * 0.005 + \frac{P\%}{N}} \text{ 最大时, } N = \frac{P\%}{0.005}$$

3、

2.8 [12/12/15] <2.2> 以下问题利用 CACTI 研究小而简单的缓存产生的影响，假定采用 65 nm (0.065 μm) 工艺。(CACTI 可以从网上获取：<http://quid.bpl.hp.com:9081/cacti/>)。

- a. [12] <2.2> 对比块大小为 64 字节的 64 KB 缓存与单组存储器的访问时间。与直接映射的组织方式相比，两路与四路组相联缓存的相对访问时间为多少？
- b. [12] <2.2> 对比块大小为 64 字节的四路组相联缓存与单组存储器的访问时间。与 16 KB 缓存相比，32 KB 与 64 KB 缓存的相对访问时间是多少？
- c. [15] <2.2> 对于 64 KB 缓存和特定工作负载，每条指令的缺失数据如下：直接映射为 0.00664、两路组相联为 0.00366、四路组相联为 0.00987、八路组相联缓存为 0.000266，求具有最短平均存储器访问时间的缓存相联度（介于 1 至 8 之间）。从整体来看，每条指令有 0.3 次数据引用。假定缓存缺失在所有模型中均耗费 10 ns。为了以周期为单位计算命中时间，假定使用 CACTI 输出周期时间，它对应于在流水线中没有气泡时，缓存的最高工作频率。

解：

a) 直接映射缓存的访问时间为 0.86ns，2 路和 4 路分别为 1.12ns 和 1.37ns。

$$2 \text{ 路相对访问时间为 } 1.12 / 0.86 = 1.30$$

$$4 \text{ 路相对访问时间为 } 1.37 / 0.86 = 1.59$$

b) 16KB 缓存的访问时间为 1.27ns，32KB 和 64KB 分别为 1.35ns 和 1.37ns。

$$32KB \text{ 的相对访问时间为 } 1.35 / 1.27 = 1.06$$

$$64KB \text{ 的相对访问时间为 } 1.37 / 1.27 = 1.078$$

c) 平均访问时间 = 命中% × 命中时间 + 未命中% × 未命中处罚，未命中% = 每条指令中错过/每条引用指令 = 2.2% (DM)、1.2% (双向)、0.33%

(4 路)，0.09% (8 路)。

直接映射访问时间= 0.86ns @ 0.5ns 周期时间= 2 周期

双向设置关联= 1.12ns @ 0.5ns 周期时间= 3 个周期

4 路集合关联= 1.37ns @ 0.83ns 周期时间= 2 个周期

8 路集关联= 2.03ns @ 0.79ns 周期时间= 3 个循环

未命中处罚=(10/0.5)= 20 周期 (DM 和 2 路) ;

$$10/0.83 = 13 \text{ 周期 (4 路)}$$

$$10/0.79 = 13 \text{ 个循环 (8 路)}$$

直接映射: $(1 - 0.022) \times 2 + 0.022 \times 20 = 2.396 \text{ 周期} \Rightarrow 2.396 \times 5 = 1.2 \text{ ns}$

2 路: $1 - 0.012 \times 3 + 0.012 \times 20 = 3.2 \text{ 周期} \Rightarrow 3.2 \times 5 = 1.6 \text{ ns}$

4 路: $(1 - 0.0033) \times 2 + 0.0033 \times 13 = 2.036 \text{ 周期} \Rightarrow 2.06 \times 0.83 = 1.69 \text{ ns}$

8 路: $(1 - 0.0009) \times 3 + 0.0009 \times 13 = 3 \text{ 周期} \Rightarrow 3 \times 0.79 = 2.37 \text{ ns}$

所以直接映射缓存是最好的选择。

4、

2.13 [10/10/10] <2.3> 考虑一个桌面系统, 它的处理器连接到一个采用纠错码 (ECC) 的 2 GB DRAM。假定只有一个宽度为 72 位的存储器通道, 其中 64 位用于数据, 8 位用于 ECC。

a. [10] <2.3> 如果使用 1 GB DRAM 芯片, DIMM 上有多少个 DRAM 芯片, 如果仅有一个 DRAM 连接到每个 DIMM 数据管脚, 每个 DRAM 必须拥有多少数据 I/O?

b. [10] <2.3> 为了支持 32 B L2 缓存块, 突发 (burst) 长度需要为多少?

c. [10] <2.3> 计算为了从某个活动页读取内容, DDR2-667 和 DDR2-533 DIMM 的峰值带宽为多少? 不计 ECC 开销。

解:

a) 带有奇偶校验或 ECC 的 2GB DRAM 实际上有 9 位字节, 需要 18 个 1Gb DRAM。

要创建 72 位输出, 每一位必须输出 $72/18 = 4$ 位

b) 突发长度为 4, 读出 32B。

c) ddr - 667 DIMM 带宽是 $667 \times 8 = 5336 \text{ MB / s}$ 。

ddr - 533 DIMM 带宽是 $533 \times 8 = 4264 \text{ MB / s}$

5、

2.18 [20/15/12] <2.3>为了从一个典型 DRAM 访问数据,必须首先激活适当的行。假定这一操作会将大小为 8 KB 的整个页面发送到行缓冲区,然后从行缓冲区中选择一个特定列。如果对 DRAM 的后续访问目标也是同一页,就可以略过激活步骤;如果不是,就必须关闭当前页,对位行进行预充电,以准备下一次激活。另一种常用 DRAM 策略是在访问结束之后立即主动关闭一个页,并对位行进行预充电。假定对 DRAM 的每次读取或写入都是采用 64 字节的大小,发送 512 位的 DDR 总线延迟(图 2-21 中的数据输出)为 T_{ddr} 。

- a. [20] <2.3>假定采用 DDR2-667,如果它需要 5 个周期进行预充电、5 个周期进行激活、4 个周期读取列,为了获得最短访问时间,如何根据行缓冲区命中率(r)选择策略? 假定对 DRAM 的每次访问之间都有足够的时间,用以完成新的随机访问。
- b. [15] <2.3>如果在对 DRAM 的所有访问中有 10%是一个接一个地发生,或者没有任何时间间隔地连续发生,应当如何改变自己的决定?
- c. [12] <2.3>使用上面计算的行缓冲区命中率,计算在采用两种策略时,每次访问的平均 DRAM 能耗差别。假定预充电需要 2 nJ,激活需要 4 nJ,从行缓冲区进行读写需要 100 pJ/位。

解:

a)

策略 1:

$$\text{预先充电延迟 } T_{rp} = 5 * \frac{1}{333MHz} = 15ns$$

$$\text{激活延迟 } T_{rcd} = 5 * \frac{1}{333MHz} = 15ns$$

$$\text{列选择延迟 } T_{cas} = 4 * \frac{1}{333MHz} = 12ns$$

$$\text{当行缓冲区被击中时的访问时间: } T_h = \frac{r(T_{cas} + T_{ddr})}{100}$$

$$\text{访问时间未击中时 } T_m = \frac{(100-r)(T_{rp} + T_{rcd} + T_{cas} + T_{ddr})}{100}$$

策略 2:

$$\text{访问时间 } T = T_{rp} + T_{cas} + T_{ddr}$$

如果 A 是总访问次数,那么当策略 1 的净访问时间等于策略 2 的总访问时间时,就会出现临界点。

$$\frac{r(T_{cas} + T_{ddr})}{100}A + \frac{(100-r)(T_{rp} + T_{rcd} + T_{cas} + T_{ddr})}{100}A = (T_{rcd} + T_{cas} + T_{ddr})A$$

$$\Rightarrow r = \frac{100T_{rp}}{T_{rp} + T_{rcd}}$$

$$r = 100 * \frac{15}{15+15} = 50\%$$

如果 r 小于 50%,那么我们必须主动关闭页面以获得最佳性能,否则我们可以保

持页面打开。

b) 关闭一个页面的关键好处是隐藏预充电延迟 T_{rp} 隐藏在关键路径。如果访问是反向的，那么这是不可能的。这个新的约束不会影响策略 1。

策略 2 的新方程，

当我们可以隐藏预充电延迟时的访问时间 $T_1 = T_{rcd} + T_{cas} + T_{ddr}$

预充电延迟处于关键路径时的接入时间 $T_2 = T_{rp} + T_{rcd} + T_{cas} + T_{ddr}$

方程现在变成：

$$\frac{r(T_{cas} + T_{ddr})}{100}A + \frac{(100 - r)(T_{rp} + T_{rcd} + T_{cas} + T_{ddr})}{100}A$$
$$= 0.9(T_{rcd} + T_{cas} + T_{ddr})A + 0.1(T_{rp} + T_{rcd} + T_{cas} + T_{ddr})A$$
$$\Rightarrow r = \frac{90T_{rp}}{T_{rp} + T_{rcd}}$$
$$r = 90 * \frac{15}{15 + 15} = 45\%$$

c) 对于任何行缓冲区命中率, 策略 2 每次访问需要额外 $r \times (2 + 4)n_j$ 。如果 $r = 50\%$ ，则策略 2 需要额外的 $3n_j$ 能量。

6、

3.1 [10]<1.8、3.1、3.2>如果在先前指令执行完毕之前，不会开始执行新的执行，那表 3-22 中代码序列的基准性能如何（用每次循环迭代的时钟周期表示）？忽略前端提取与译码过程。假定执行进程没有因为缺少下一条指令而停顿，但每个周期只能发射一条指令。假定该分支被选中，而且存在一个时钟周期的分支延迟槽。

表3-22 练习3.1至练习3.6的代码与延迟				
			超过一个时钟周期的延迟	
Loop:	LD	F2,0(RX)	存储器LD	+4
I0:	DIVD	F8,F2,F0	存储器SD	+1
I1:	MULTD	F2,F6,F2	整数ADD, SUB	+0
I2:	LD	F4,0(Ry)	分支	+1
I3:	ADD0	F4,F0,F4	ADD0	+1
I4:	ADD0	F10,F8,F2	MULTD	+5
I5:	ADDI	Rx,Rx,#8	DIVD	+12
I6:	ADDI	Ry,Ry,#8		
I7:	SD	F4,0(Ry)		
I8:	SUB	R20,R4,Rx		
I9:	BNZ	R20,Loop		

解：

Loop:	LD	F2,0(Rx)	1+4
	DIVD	F8,F2,F0	1+12
	MULTD	F2,F6,F2	1+5
	LD	F4,0(Ry)	1+4
	ADDD	F4.F0,F4	1+1
	ADDD	F10,F8,F2	1+1
	ADDI	Rx,Rx,#8	1
	ADDI	Ry,Ry,#8	1
	SD	F4,0(Ry)	1+1
	SUB	R20,R4,Rx	1
	BNZ	R20,Loop	<u>1+1</u>
	每个循环周期		40

表 3-22 中代码序列的基准性能(以循环为单位,每次循环迭代)为 40,如果在第一条指令执行完成之前不能启动新指令的执行,每条指令都需要一个执行的时钟周期(一个时钟周期,在这个时钟周期中,该指令,而且只有该指令,占用执行单元;因为每条指令都必须执行,所以这个循环至少需要那么多的时钟周期),则为 40。

7、

3.9 [5] <3.4>如果你不清楚寄存器重命名程序必须做哪些工作,可以回过头来看看正在执行的汇编代码,问问自己,必须具备哪些条件才能获得正确结果。例如,考虑一个 3 路超标量机器,同时对下面这 3 条指令进行重命名:

```
ADDI R1, R1, R1
ADDI R1, R1, R1
ADDI R1, R1, R1
```

如果 R1 的值在初始为 5,那么执行这一序列之后,它的值应当是多少?

解:在执行这个序列之后, R1 的值为 40.

ADDI	R1, R1, R1	5+5 → 10
ADDI	R1, R1, R1	10+10 → 20
ADDI	R1, R1, R1	20+20 → 40

8、

4.5 [25] <4.4>现在假定我们希望使用单个线程块在 GPU 上实施 MrBayes 内核。使用 CUDA 改写内核的 C 代码。假定指向条件似然表和转移概率表的指令以内核参数的形式指定。为循环的每个迭代调用一个线程。对于任何需要重复使用的值,应当先将其载入共享存储器中,然后再进行操作。

解:

```
__global__ void compute_condLike (float *clL, float *clR, float *clP, float
```

```

*tiPL, float *tiPR) {
    int i,k = threadIdx.x;
    __shared__ float clL_s[4], clR_s[4];
    for (i=0;i<4;i++) {
        clL_s[i]=clL[k*4+i];
        clR_s[i]=clR[k*4+i];
    }

    clP[k*4] = (tiPL[k*16+AA]*clL_s[A] + tiPL[k*16+AC]*clL_s[C] +
tiPL[k*16+AG]*clL_s[G] + tiPL[k*16+AT]*clL_s[T])*(tiPR[k*16+AA]*clR_s[A] +
tiPR[k*16+AC]*clR_s[C] + tiPR[k*16+AG]*clR_s[G] + tiPR[k*16+AT]*clR_s[T]);

    clP[k*4+1] = (tiPL[k*16+CA]*clL_s[A] + tiPL[k*16+CC]*clL_s[C] +
tiPL[k*16+CG]*clL_s[G] + tiPL[k*16+CT]*clL_s[T])*(tiPR[k*16+CA]*clR_s[A] +
tiPR[k*16+CC]*clR_s[C] + tiPR[k*16+CG]*clR_s[G] + tiPR[k*16+CT]*clR_s[T]);

    clP[k*4+2] = (tiPL[k*16+GA]*clL_s[A] + tiPL[k*16+GC]*clL_s[C] +
tiPL[k*16+GG]*clL_s[G] + tiPL[k*16+GT]*clL_s[T])*(tiPR[k*16+GA]*clR_s[A] +
tiPR[k*16+GC]*clR_s[C] + tiPR[k*16+GG]*clR_s[G] + tiPR[k*16+GT]*clR_s[T]);

    clP[k*4+3] = (tiPL[k*16+TA]*clL_s[A] + tiPL[k*16+TC]*clL_s[C] +
tiPL[k*16+TG]*clL_s[G] + tiPL[k*16+TT]*clL_s[T])*(tiPR[k*16+TA]*clR_s[A] +
tiPR[k*16+TC]*clR_s[C] + tiPR[k*16+TG]*clR_s[G] + tiPR[k*16+TT]*clR_s[T]);
}

```

9、

4.13 [10/15] <4.4>假定有一种包含 10 个 SIMD 处理器的 GPU 体系结构。每条 SIMD 指令的宽度为 32，每个 SIMD 处理器包含 8 个车道，用于执行单精度运算和载入/存储指令，也就是说，每个非分岔 SIMD 指令每 4 个时钟周期可以生成 32 个结果。假定内核的分岔分支将导致平均 80%的线程为活动的。假定在所执行的全部 SIMD 指令中，70%为单精度运算、20%为载入/存储。由于并不包含所有存储器延迟，所以假定 SIMD 指令平均发射率为 0.85。假定 GPU 的时

钟速度为 1.5 GHz。

a. [10] <4.4> 计算这个内核在这个 GPU 上的吞吐量，单位为 GFLOP/s。

b. [15] <4.4> 假定我们有以下选项：

(1) 将单精度车道数增大至 16。

(2) 将 SIMD 处理器数增大至 15（假定这一改变不会影响所有其他性能度量，代码会扩展到增加的处理器上）。

(3) 添加缓存可以有效地将存储器延迟缩减 40%，这样会将指令发射率增加至 0.95，对于这些改进中的每一项。

吞吐量的加速比为多少？

解：

$$a) 1.5\text{GHz} \times 0.80 \times 0.85 \times 0.70 \times 10\text{cores} \times 32 \div 4 = 57.12 \text{ G FLOPs/s}$$

$$b) (1) 1.5\text{GHz} \times 0.80 \times 0.85 \times 0.70 \times 10\text{cores} \times 32 \div 2 = 114.24 \text{ G FLOPs/s}$$

$$\text{加速比: } \frac{114.24}{57.12} = 2$$

$$(2) 1.5\text{GHz} \times 0.80 \times 0.85 \times 0.70 \times 15\text{cores} \times 32 \div 4 = 85.68 \text{ G FLOPs/s}$$

$$\text{加速比: } \frac{85.68}{57.12} = 1.5$$

$$(3) 1.5\text{GHz} \times 0.80 \times 0.95 \times 0.70 \times 10\text{cores} \times 32 \div 4 = 63.84 \text{ G FLOPs/s}$$

$$\text{加速比: } \frac{63.84}{57.12} = 1.11$$