

1.在为服务器供电时，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果影响一个系统的必要冷却，从而影响它的价格。请使用表进行功率计算

- A. 机架上安装一个冷却门的成本是 4000 美元，可以消散 14KW（向室内散热，向室外散热需要增加成本）。如果服务器采用奔腾 4 处理器，1GB240 管脚 DRAM 和一个 7200rpm 硬盘驱动器，那么一个冷却门可以冷却多少个服务器？
- B. 我们正在考虑为硬盘驱动器提供容错功能。RAID1 将磁盘数目加倍，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. 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，在一个机架内可以放入多少个 a 中提到的服务器？需要多少个冷却门？

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：

$$A. 14KW / (66W + 2.3W + 7.9W) = 183.7 = 183$$

$$B. 14KW / (66W + 2.3W + 2 * 7.9W) = 166.4 = 166$$

$$C. 200W * 11 = 2200W$$

$$2200W / 66W + 2.3W + 7.9W = 28$$

一个机架可以放入 28 个 a 中的服务器，28 < 183

所以只需要一个冷却门

2.公司刚刚购买了一个新的 Intel Core i5 双核处理器，你接到针对这一处理器来优化软件的任务。你将在这个双核处理器上运行两个应用程序，但他们的资源需求并不一样，第一个程序需要 80% 的资源，另一个仅需要 20% 的资源。假定对该程序的一部分进行并行化，该部分的加速比为 2.

- A. 假定第一个应用程序的 40% 可以并行化，那么在隔离运行时，通过这个应用程序可以实现多大的加速比？
- B. 假定第二个应用程序的 99% 可以并行化，那么在隔离运行时，通过这个应用程序可以实现多大的加速比？
- C. 假定第一个应用程序的 40% 可以并行化，如果对其实现并行化，系统总加速比为多少？
- D. 假定第二个应用程序的 99% 可以并行化，如果对其实现并行化，系统总加速比为多少？

解：

$$A. \text{总加速比} = 1 / (1 - \text{升级比例}) + \text{升级比例} / \text{升级加速比} \quad \text{即 } 1 / (1 - 0.4 + 0.4 / 2) = 1.25$$

$$B. \text{总加速比} = 1 / (1 - \text{升级比例}) + \text{升级比例} / \text{升级加速比} \quad \text{即 } 1 / (1 - 0.99 + 0.99 / 2) = 1.98$$

$$C. \text{总加速比} = 1 / (1 - \text{升级比例}) + \text{升级比例} / \text{升级加速比} \\ \text{即 } 1 / (0.2 + 0.8 * (1 - 0.4 + 0.4 / 2)) = 1 / (0.2 + 0.64) = 1.19$$

$$D. \text{总加速比} = 1 / (1 - \text{升级比例}) + \text{升级比例} / \text{升级加速比} \\ \text{即 } 1 / (0.8 + 0.2 * (1 - 0.99 + 0.99 / 2)) = 1 / (0.8 + 0.101) = 1.11$$

3. 使用图 2-21 中的程序结果示例回答一下问题。
- 第二级缓存的总大小和块大小为多少？
  - 第二级缓存的缺失代价为多少？
  - 第二级缓存的相连度为多少？
  - 主存储器的大小是多少？
  - 如果页大小为 4KB，而分页时间为多少？

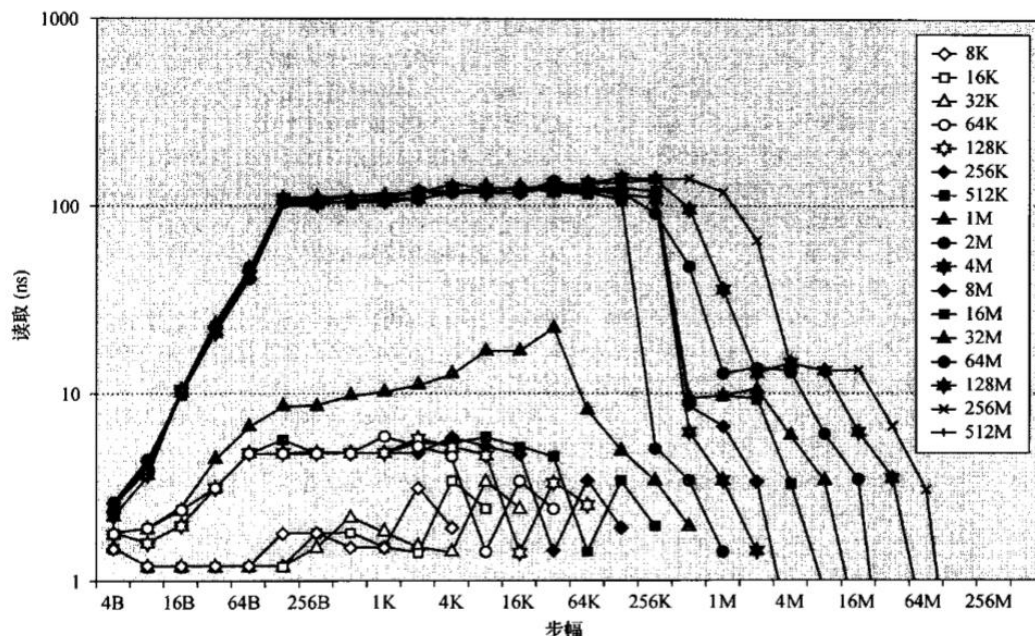


图 2-21 代码清单 2-1 中程序的输出举例

解：

- 根据圆形结点的曲线，第二级缓存总大小为 1MB，块大小为 128B。
- 根据圆形结点的曲线的转折点，第二级缓存的缺失代价为 110ns。
- 第二级缓存的相联度为  $1024/128=8$ ，八路相联。
- 代表 512M 的图线并没有出现，所以主存储器大小为 512MB。
- 当以 16B 的步幅浏览页表时，每次引用需要 946ns。对于每页  $4KB/16B=256$  个这样的引用，结果大约是每页 242ms。

4. 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 2GB DRAM。假定只有一个宽度为 7 位的存储器通道，其中 64 位用于数据，8 位用于 ECC。

- 如果使用 1GB 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？
- 为了支持 32BL2 缓存块，突发长度需要多少？
- 计算为了从某个活动页读取内容，DDR2-667 和 DDR2-533DIMM 的峰值带宽为多少？不计 ECC 开销。

解：

- 对于桌面系统和服务器系统 DIMM 通常被组织为 8 字节宽度（+ECC），故一个采用 ECC 或者奇偶校验位的 2GB DRAM 为 9 字节宽度，需要由 18 个 1GB DRAM 芯片来组成，为了产生  $9 \times 8=72$  位的 I/O，每个 DRAM 需要有  $72/18=4$  位数据 I/O。
- 突发长为  $2^7/32=4$ 。

C.

标 准	时钟频率 (MHz)	每秒传输个数 (百万个)	DRAM名称	MB/s/DIMM	DIMM名称
DDR	133	266	DDR266	2128	PC2100
DDR	150	300	DDR300	2400	PC2400
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10 664	PC10700
DDR3	800	1600	DDR3-1600	12 800	PC12800
DDR4	1066-1600	2133-3200	DDR4-3200	17 056-25 600	PC25600

DDR2-667DIMM 峰值带宽为  $667 \times 8 = 5336 \text{ MB/s}$

DDR2-533DIMM 峰值带宽为  $533 \times 8 = 4264 \text{ MB/s}$

5.假设各种分支指令数占有指令数的百分比如下:

条件分支	20% (其中的 60%是分支成功的)
跳转和调用	5%

现有一条段数为 4 的流水线, 无条件分支在第二个时钟周期结束时就被解析出来, 而条件分支要到第三个时钟周期结束时才能被解析出来。第一个流水段是完全独立于指令类型的, 即所有类型的指令都必须经过第一个流水段的处理。请问在没有任何控制相关的情况下, 该流水线相对于存在上述控制相关情况下的加速比是多少?

解: 没有控制相关时流水线的平均  $\text{CPI}=1$

存在控制相关是: 由于无条件分支在第二时钟周期结束是就被解析出来, 而条件分支要到第三个时钟周期结束时才能被解析出来。所以:

(1) 若使用排空流水线策略, 则对于条件分支, 有两个额外的停顿周期, 对于无条件分支, 有一个额外的停顿周期:

$$\text{CPI}=1+20\% \times 2+5\% \times 1=1.45$$

$$\text{加速比 } S=\text{CPI}/1=1.45$$

(2) 若使用预测分支成功策略, 则对于不成功的条件分支, 有两个额外的停顿周期, 对于无条件分支和成功的条件分支, 有一个额外的停顿周期:

$$\text{CPI}=1+20\% \times (60\% \times 1+40\% \times 2)+5\% \times 1=1.33$$

$$\text{加速比 } S=1.33$$

(3) 若使用预测分支失败策略, 则对于成功的条件分支, 有两个额外的停顿周期, 对于无条件分支, 有一个额外的停顿周期, 对于不成功的条件分支, 其目标地址已经由 PC 值给出, 不必等待, 所以无延迟:

$$\text{CPI}=1+20\% \times (60\% \times 2+40\% \times 0)+5\% \times 1=1.29$$

$$\text{加速比 } S=1.29$$

6.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 *cL,float *cR,float *cIP,float *tiPL,float *tiPR) {int i,k=threadIdx.x;__shared__ float cL_s[4], cR_s[4];
```

```

for (i=0;i<4;i++) {cIL_s[i]=cIL[k*4+i]; cIR_s[i]=cIR[k*4+i];}
cIP[k*4]=(tiPL[k*16+AA]*cIL_s[A]+tiPL[k*16+AC]*cIL_s[C]+tiPL[k*16+AG]*cIL_s[G]+tiPL[k*16+AT]*cIL_s[T])*(tiPR[k*16+AA]*cIR_s[A]+tiPR[k*16+AC]*cIR_s[C]+tiPR[k*16+AG]*cIR_s[G]+tiPR[k*16+AT]*cIR_s[T]);
cIP[k*4+1]=(tiPL[k*16+CA]*cIL_s[A]+tiPL[k*16+CC]*cIL_s[C]+tiPL[k*16+CG]*cIL_s[G]+tiPL[k*16+CT]*cIL_s[T])*(tiPR[k*16+CA]*cIR_s[A]+tiPR[k*16+CC]*cIR_s[C]+tiPR[k*16+CG]*cIR_s[G]+tiPR[k*16+CT]*cIR_s[T]);
cIP[k*4+2]=(tiPL[k*16+GA]*cIL_s[A]+tiPL[k*16+GC]*cIL_s[C]+tiPL[k*16+GG]*cIL_s[G]+tiPL[k*16+GT]*cIL_s[T])*(tiPR[k*16+GA]*cIR_s[A]+tiPR[k*16+GC]*cIR_s[C]+tiPR[k*16+GG]*cIR_s[G]+tiPR[k*16+GT]*cIR_s[T]);
cIP[k*4+3]=(tiPL[k*16+TA]*cIL_s[A]+tiPL[k*16+TC]*cIL_s[C]+tiPL[k*16+TG]*cIL_s[G]+tiPL[k*16+TT]*cIL_s[T])*(tiPR[k*16+TA]*cIR_s[A]+tiPR[k*16+TC]*cIR_s[C]+tiPR[k*16+TG]*cIR_s[G]+tiPR[k*16+TT]*cIR_s[T]); }

```

7.假定图 5-24 的缓存内容和表 5-12 中实现方式 1 的定时参数。以下代码序列在基本协议和练习 5.5 的新 MOSI 协议中的总停顿周期为多少？假定不需要互联事务的状态转换不会导致额外停顿周期。

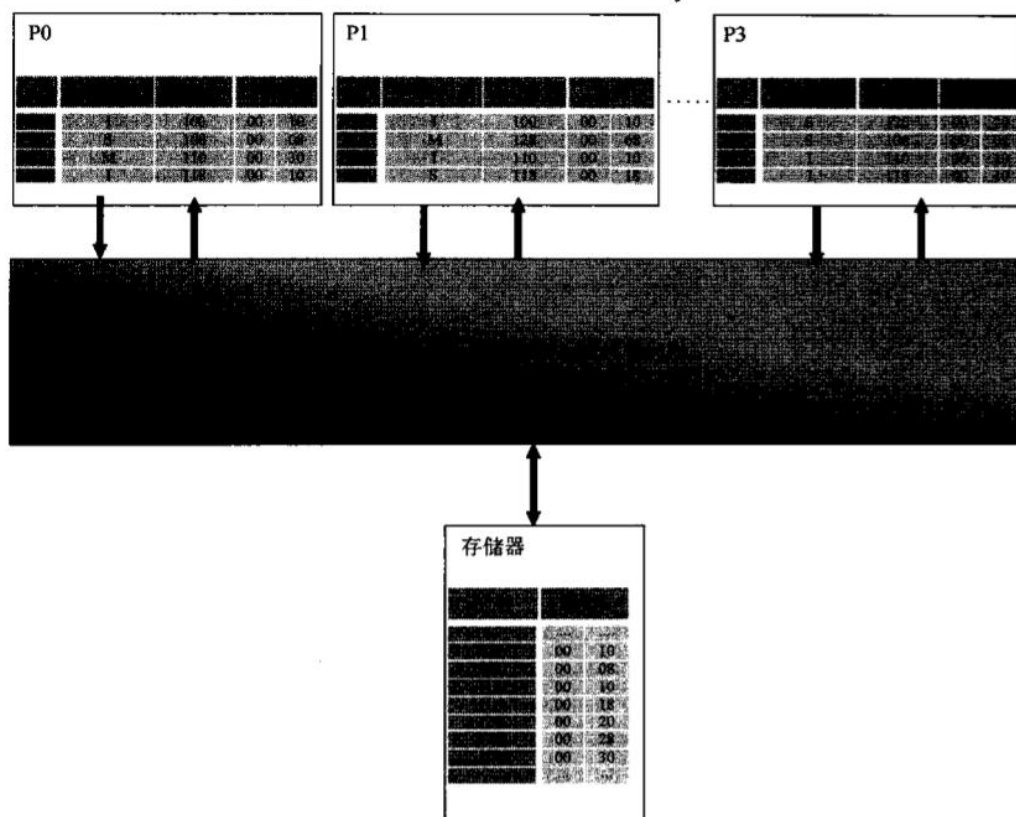


图 5-24 多核（点对点）多处理器

表5-12 监听一致性延迟

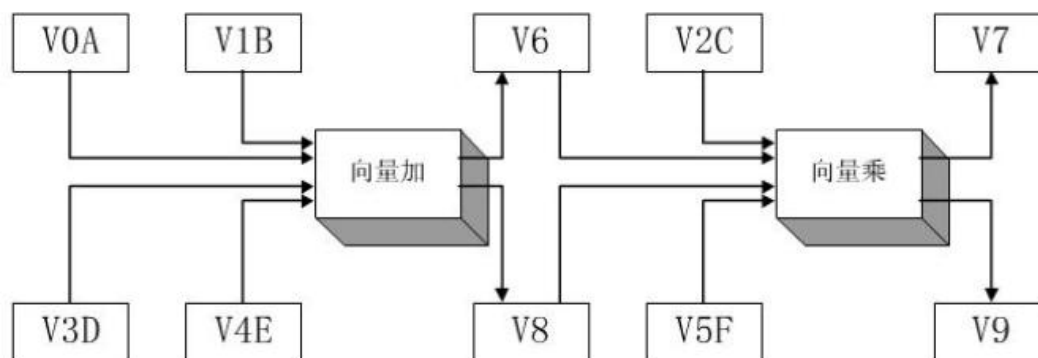
参 数	实现方式1	实现方式2
N <sub>存储器</sub>	100	100
N <sub>缓存</sub>	40	130
N <sub>失效</sub>	15	15
N <sub>写回</sub>	10	10

- A. P0:read 100  
P0:write 100 <--40
- B. P0:read 120  
P0:write 120 <--60
- C. P0:read 100  
P0:write 120
- D. P0:read 100  
P1:write 100 <--60
- E. P0:read 100  
P0:write 100 <--60  
P0:write 100 <--40

解:

- A. P1: read 110, 读不命中, P0 的缓存  
P3: read 110, 读不命中, MSI 到达内存, MOSI 到达 P0 的缓存  
P0: read 110, 读命中  
MSI:  $40 + 10 + 100 + 0 = 150$  停顿周期  
MOSI:  $40 + 10 + 40 + 10 + 0 = 100$  停顿周期
- B. P1: read 120, 读不命中, 到达内存  
P3: read 120, 读命中  
P0: read 120, 读不命中, 到达内存  
两个协议相等:  $100 + 0 + 100 = 200$  个停顿周期
- C. P0: write 120 <-- 80, 写不命中, 使 P3 无效  
P3: read 120, 读不命中, P0 的缓存  
P0: read 120, 读命中  
两个协议相等:  $100 + 40 + 10 + 0 = 150$  停顿周期
- D. P0: write 108 <-- -88, 发送无效, 使 P3 无效  
P3: read 108, 读不命中, P0 的缓存  
P0: write 108 <-- -98, 发送无效, 使 P3 无效  
两个协议相等:  $15 + 40 + 10 + 15 = 80$  个停顿周期

8. 某向量处理机有 16 个向量寄存器, 其中 V0-V5 分别放有向量 A、B、C、D、E、F, 向量长度均为 8, 向量各元素均为浮点数; 处理部件采用两条单功能流水线, 加法功能部件时间为 2 拍, 乘法功能部件时间为 3 拍。采用类似于 CARY-1 的链接技术, 先计算  $(A+B)*C$ , 在流水线不停流的情况下, 接着计算  $(D+E)*F$



(1) 求此流水线的通过时间 (设寄存器入、出各需 1 拍)

(2) 假如每拍时间为 50ns, 完成这些计算并把结果存进相应寄存器, 此处理部件的实际吞吐量是多少 MFLOPS?

解: (1) 假设  $A+B$  的中间结果放在  $V6$  中,  $(A+B)*C$  的最后结果放在  $V7$  中,  $D+E$  的中间结果放在  $V8$  中,  $(D+E)*F$  的最后结果放在  $V9$  中。通过时间应该为  $(A+B)*C$  通过的时间  $T_{通过} = (1+2+1) + (1+3+1) = 9$  拍

(2) 在做完  $(A+B)*C$  之后, 作  $(C+D)*E$  就不需要通过时间了。

$V6 \leftarrow A+B$

$V7 \leftarrow V6 * C$

$V8 \leftarrow D+E$

$V9 \leftarrow V8 * F$

$T = T_{通过} + (8-1) + 8 = 24$  拍  $= 24 * 50 = 1200ns$

$TP = 32 / T = 32 / 1200 = 26.67MFLOPS$

9. 一个具有 32 台处理机的系统, 对远程存储器访问时间是 2000ns。除了通信以外, 假设计算中的访问均命中局部存储器。当发出一个远程请求时, 本地处理机挂起。处理机的时钟周期时间是 10ns, 假设指令基本的 CPI 为 1.0 (设所有访存均命中 Cache)。对于下述两种情况:

(1) 没有远程访问

(2) 0.5% 的指令需要远程访问

解: 已知远程访问率  $p=0.5\%$ , 远程访问时间  $t=2000ns$ , 时钟周期  $T=10ns$

远程访问开销  $C = t/T = 2000ns/10ns = 200$  (时钟周期数)

有 0.5% 远程访问的机器的实际 CPI2 为:

$CPI2 = CPI1 + P * C = 1.0 + 0.5\% * 200 = 2.0$

只有局部访问的机器的基本  $CPI1=1.0$

$CPI2/CPI1 = 2.0/1.0 = 2$  倍

10. 给定一下的假设, 试计算直接映像 Cache 和两路组相连 Cache 的平均访问时间以及 CPU 的性能。由计算结果能得出什么结论?

(1) 理想 Cache 情况下 CPI 为 2.0, 时钟周期为 2ns, 平均每条指令访存 1.2 次

(2) 两者 Cache 容量均为 64KB, 块大小都是 32 字节

(3) 组相连 Cache 中的多路选择器使 CPU 的时钟周期增加了 10%

(4) 这两种 Cache 的失效开销都是 80ns

(5) 命中时间为 1 个时钟周期

(6) 64KB 直接映像 Cache 的失效率为 1.4%，64KB 两路组相连 Cache 的失效率为 1.0%

解：平均访问时间=命中时间+失效率\*失效开销

平均访问时间 1-路= $2.0 + 1.4\% \times 80 = 3.12\text{ns}$

平均访问时间 2-路= $2.0 \times (1 + 10\%) + 1.0\% \times 80 = 3\text{ns}$

故两路组相连的平均访问时间比较低

$\text{CPUtime} = (\text{CPU 执行} + \text{存储等待周期}) \times \text{时钟周期}$

$\text{CPUtime} = \text{IC} (\text{CPI 执行} + \text{总失效次数/指令总数} \times \text{失效开销}) \times \text{时钟周期} =$

$\text{IC} ((\text{CPI 执行} \times \text{时钟周期}) + (\text{每条指令的访存次数} \times \text{失效率} \times \text{失效开销} \times \text{时钟周期}))$

$\text{CPUtime 1-路} = \text{IC} (2.0 \times 2 + 1.2 \times 0.014 \times 80) = 5.344\text{IC}$

$\text{CPUtime 2-路} = \text{IC} (2.2 \times 2 + 1.2 \times 0.01 \times 80) = 5.36\text{IC}$

相对性能比： $\text{CPUtime 1-路} / \text{CPUtime 2-路} = 5.36 / 5.344 = 1.003$

直接映像 Cache 的访问速度比两路组相连 Cache 要快 1.04 倍，而两路组相连 Cache 的平均性能比直接映像 Cache 要高 1.003 倍。