高等计算机体系结构，2019 年春季

作业 5： Cache 和 Memory（参考答案）

主讲教师：栾钟治

助讲教师：杨海龙；助教：许崇杨，左佩璇

作业下发时间：2019 年 5 月 06 日

作业回收时间：2019 年 5 月 27 日

1 Cache 10 分

下面给出了运行在带数据 cache 的处理器上的程序所生成的四种不同的地址序列，同时给出了每种序列的 cache 命中率。假设 cache 在每个序列开始时是空的，请回答该处理器数据 cache 的下述参数分别是多少：

(a) 相联度(1, 2 还是 4 路)

参考答案：

4

对于序列 2, 块 0, 512, 1024 和 1536 是仅有的重用块，也就是会第二次访问并可能会导致 cache 命中的块，其中 3 块应该在第二次被访问时命中，因此命中率才会是 0.33 (3/9)。

块大小是 8 字节(见下一问)，对于任何的 cache 大小 (256B 或 512B)，这些块都映射到 set 0。因此，相联度是 1 或者 2 会造成四块中最多 1 或 2 块在第二次访问时在 cache 中，使得最大的可能命中率小于 3/9，而这个序列的命中率是 3/9，说明相联度只能是 4。

(b) 块大小(1, 2, 4, 8, 16 还是 32 字节)

参考答案：

8 字节

对于序列 1， 6 次访问中只有 2 次(地址 2 和 4) 能够 cache 命中，命中率是 0.33。除了 8 字节外其他的块大小都不能满足命中率为 0.33，要么大要么小。

(c) cache 总容量(256 还是 512 字节)

参考答案：

256 字节

对于序列 3，512 字节的容量会使命中率达到 4/9(4 路组相联，8 字节 cache 块，无论什么替换策略)，高于 0.33，所以 cache 总容量是 256 字节。

(d) 替换策略(LRU 还是 FIFO)

参考答案：

LRU

对于上述的参数，所有序列 4 中的 cache 块被映射到组 0，如果使用 FIFO 替换策略，命中率是 3/8，而采用 LRU 替换策略命中率是 1/4，所以替换策略是 LRU。

假设：所有的访存都是单字节的访问，所有的地址都是字节地址。

序列	地址序列	命中率
1	0, 2, 4, 8, 16, 32	0.33
2	0, 512, 1024, 1536, 2048, 1536, 1024, 512, 0	0.33
3	0, 64, 128, 256, 512, 256, 128, 64, 0	0.33
4	0, 512, 1024, 0, 1536, 0, 2048, 512	0.25

2 内存的交叉存取 20 分

2.1 一台机器有 4 KB 的主存，由 1 个通道、1 个 rank 和 N(N>1)个 bank 构成。系统没有虚拟存储。

1) 数据采用 cache 块交叉存取策略进行交叉存取，即连续的 cache 块对应到连续的 bank 上；
2) cache 块大小为 32 字节，bank 的 1 行有 128 字节；
3) 采用打开行策略，即行缓冲中的行在被访问后继续保持在行缓冲中，直到有别的行被访问；
4) 行缓冲命中指访问的行存在于行缓冲中，行缓冲缺失指访问的行不在行缓冲中。

(a) 某个程序在这台机器上执行，访问以下字节时(数字表示字节的位置，比如 320 表示第 320 个字节)发生片上 cache 缺失而需要访存：0, 32, 320, 480, 4, 36, 324, 484, 8, 40, 328, 488, 12, 44, 332, 492，若行缓冲命中率为 0，即所有访问的行都不在行缓冲中，请问 bank 数 N 的最小值是多少?

参考答案：
2 个
Cache 块大小是 32 字节，所以，对于给定的访存序列相应的 cache 块访问序列是 0, 1, 10, 15, 0, 1, 10 ,15, 0, 1, 10, 15, 0, 1, 10, 15。
当 bank 数是 1 时，所有 cache 块映射到同一个 bank，块 0、1、10 和 15 映射到行 0、0、2 和 3。所以， 当块 1 紧接着块 0 被访问时，会产生行缓冲命中，即行缓冲命中率不为 0。
当 bank 数是 2 时，块 0、1、10 和 15 映射到不同的行和 bank (bank 0, 行 0；bank 1, 行 0；bank 0, 行 1；bank 1, 行 1)。这样，访问序列就是(bank 0, 行 0), (bank 1, 行 0), (bank 0, 行 1), (bank 1, 行 1) (重复四次)。
因此，每个 bank 上的行 0 和 1 被交替访问，导致行缓冲命中率为 0。

(b) 如果对于同一个序列，行缓冲命中率是 75%，请问 bank 数 N 的最小值是多少?

参考答案：
4 个
当 bank 数是 1 时, cache 块 0、1、10 和 15 映射到行 0、0、2 和 3 (与 a 中相同)。这时的行访问序列为 0, 0, 2, 3 (重复四次)，其中 3 次访问行缓冲不命中，命中率是 25%。
对于其它数量的 bank, 块 0、1、10 和 15 映射到不同的行。
假设有这四个 cache 块的行没有在行缓冲中打开，那么最大的命中率只能是 75%，因为对每个块的第一次访问不命中(强制缺失)。这个最大命中率(75%)意味着每个块除了第一次访问后续访问不能有不命中，因此，含有四个块的行必须映射到不同的 bank，即最少有 4 个 bank。

(c) i) 对于同一序列，行缓冲的命中率能达到 100%吗? 请解释原因

参考答案：

四个 cache 块映射到不同的行，因此行缓冲命中率达到 100%的唯一可能就是包含每个块的行都已经在行缓冲中打开。

ii) 如果能达到, 最少需要多少 bank 才能够获得 100%的行缓冲命中率?

参考答案：
4 个 bank 就足以实现，只要 4 个分别包含 4 个 cache 块的行都已经打开(分别在 4 个 bank)。

2.2 一个 DRAM 主存储系统由 1 个通道、1 个 rank 和 N 个 bank 构成。Bank 一行 256 字节，一个 cache 块 64 字节。数据采用跨 bank 的行交叉存取方式组织，物理地址的分配方案如下：

行	Bank	列	BiB

采用打开行策略，即行缓冲中的行在被访问后继续保持在行缓冲中，直到有别的行被访问。初始时，所有 bank 的第 1024 行打开。
(a) 当有如下的 cache 块访问序列时，如果系统的行缓冲命中率为 33.3% (即 1/3)，请问系统中共有多少个 bank:
0, 4, 8, 16, 32, 64, 128, 256, 128, 64, 32, 16, 8, 4, 0

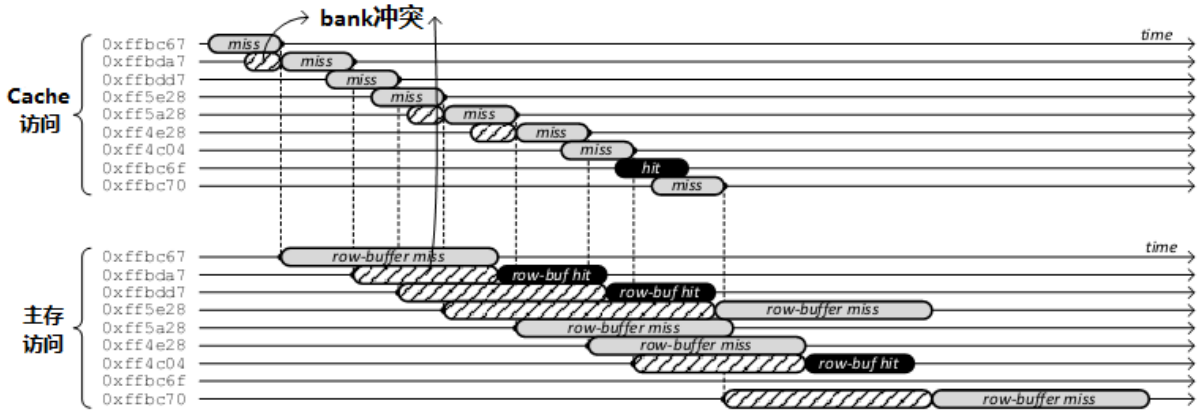
参考答案：
2^5= 32 bank

(b) 如果行缓冲命中率是 7/15，请问系统中共有多少个 bank?

参考答案：
2^7= 128 bank

3 Bank 25 分

一个处理器的分层存储结构由一个小的 SRAM L1-cache 和一个大的 DRAM 主存储器组成，SRAM 和 DRAM 都被划分成 bank。处理器有 24 位物理地址空间，并且不支持虚拟存储(即所有地址都是物理地址)。某个应用开始在这个处理器上运行，下图显示了在时间尺度上应用对存储系统引用的过程(包括在 L1-cache 和主存中)。
例如, 应用对存储第一次引用的字节地址是 0xffbc67(假设所有的引用都是对按字节编址的内存地址的按字节读取)，但是这次引用在 L1-cache 中不命中 (假设 L1-cache 初始时为空)。紧接着，应用访问主存，这会经历一次行缓冲的不命中(初始时，假设主存的所有 bank 都打开一个永远不会被任何应用访问的行)。最后，包含字节地址 0xffbc67 的 cache 块从主存取到 cache 中。
随后的内存引用可能会经历 L1-cache 和/或主存的 bank 冲突(当某个特定的 bank 还在提供之前的某个引用时)。



下表用 16 进制和 2 进制分别给出了该应用对存储系统引用的地址序列。

16 进制	2 进制
ffbc67	1111 1111 1011 1100 0110 0111
ffbda7	1111 1111 1011 1101 1010 0111
ffbdd7	1111 1111 1011 1101 1101 0111
ff5e28	1111 1111 0101 1110 0010 1000
ff5a28	1111 1111 0101 1010 0010 1000
ff4e28	1111 1111 0100 1110 0010 1000
ff4c04	1111 1111 0100 1100 0000 0100
ffbc6f	1111 1111 1011 1100 0110 1111
ffbc70	1111 1111 1011 1100 0111 0000

请分析上面的图和表，回答下列有关处理器上 cache 和主存的组织相关的问题，以下是一些假设：
1) L1-cache 的假设
块大小: ? (2 的幂, 大于 2)
相联度: ? (2 的幂, 大于 2)
数据存储的大小: ? (2 的幂, 大于 2)
Bank 数: ? (2 的幂, 大于 2)
初始时为空
2) 主存的假设
通道数: 1
每通道 rank 数: 1
每 rank 的 bank 数: ? (2 的幂, 大于 2)
没 bank 的行数: ? (2 的幂, 大于 2)
每行的 cache 块数: ? (2 的幂, 大于 2)
包含应用的整个工作集
初始时，所有的 bank 打开第 0 行，应用永远不会访问该行

注意: 对于以下问题，假设所有的偏移量和索引来自连续的地址位
(a) L1-cache 的块大小是多少字节? 24-bit 物理地址中哪些位是 cache 块偏移量? (物理地址的最低位为 0 位)

参考答案：
块大小: 16 字节

块偏移量所在位置: 0-3 位

(b) L1-cache 有多少 bank? 24-bit 物理地址中哪些位是 L1-cache 的 bank 索引? (物理地址的最低位为 0 位)

参考答案:
L1-cache 的 bank 数: 4
L1-cache bank 索引位的位置: 4-5

(c) 主存中有多少 bank? 24-bit 物理地址中哪些位是主存的 bank 索引? (物理地址的最低位为 0 位)

参考答案:
主存 bank 数: 8
主存 bank 索引位的位置: 10-12

(d) 物理地址向主存映射时用了什么样的交叉存取方案?

参考答案:
行交叉存取

(e) 为了支持 24-bit 的物理地址空间，主存的每一个 bank 需要多少行? 24-bit 物理地址中哪些位是主存的行索引? (物理地址的最低位为 0 位)

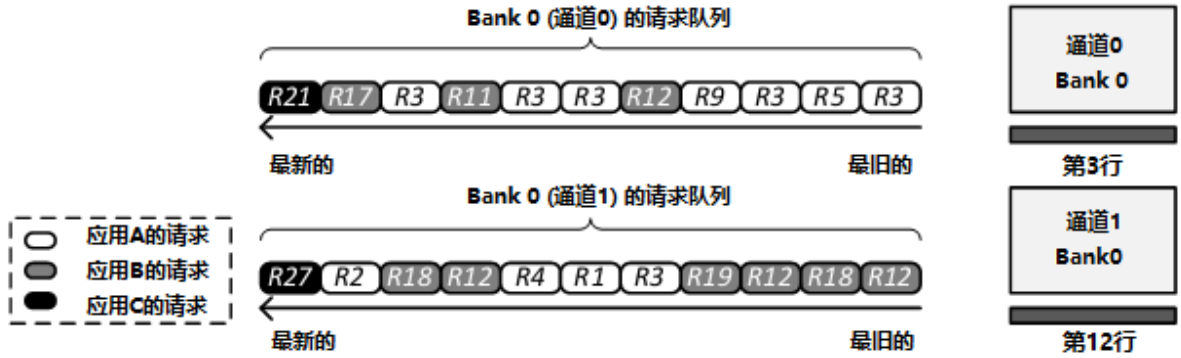
参考答案:
每个主存 bank 的行数: 2048
行索引位的位置: 13-23

(f) 在一行中的每个 cache 块被称为列，一行中有多少列? 24-bit 物理地址中哪些位是主存的列索引? (物理地址的最低位为 0 位)

参考答案:
每行的列数: 64
列索引位的位置: 4

4 内存调度 25 分

为了响应访存请求，内存控制器会发射 1 条或多条 DRAM 命令以从 bank 访问数据。有 4 种不同的 DRAM 命令。
1) 激活(ACTIVATE): 取被访问的行装入 bank 的行缓冲。 这一操作也被称为打开行(延迟: 15ns)
2) 预充电(PRECHARGE): 将 bank 的行缓冲中的内容存回行。这一操作也被称为关闭行 (延迟: 15ns)
3) 读/写: 从行缓冲中访问数据(延迟: 15ns)
下图显示了在时刻 t_0 时内存控制器中的内存请求缓冲的快照。每一个请求按照颜色的不同代表了其所属的不同应用(假设所有的应用运行在独立的核上)。同时，每个请求标注了它要访问的行地址(或索引)，例如 R3 表示请求的是第 3 行。另外，假设所有的请求都是读请求。



访存请求在读命令完成后被响应(即读命令被发射 15ns 之后)，每个应用(A、B 或 C)停顿直到它所有访存请求被响应为止。

假设初始时(t_0 时)，每个 bank 的第 3 行和第 12 行分别被取出并存入行缓冲，没有任何其他应用的请求到达内存控制器。

4.1 非应用感知的调度策略

(a) 使用先来先服务调度策略(FCFS)，每个应用的停顿时间是多少?

参考答案：

H-命中的延迟，M-缺失的延迟

应用 A: MAX (2H+7M, H+9M) = H+9M = 15+405 = 420ns

应用 B: MAX (2H+8M, H+8M) = 2H+8M = 30+360 = 390ns

应用 C: MAX (2H+9M, H+10M) = H+10M = 15+450 = 465ns

(b) 使用行缓冲优先加先来先服务的调度策略(FR-FCFS)，每个应用的停顿时间是多少?

参考答案：

应用 A: MAX (5H+2M, (4H+2M)+4M) = 4H+6M = 60+270 = 330ns

应用 B: MAX ((5H+2M)+3M, 4H+2M) = 5H+5M = 75+225 = 300ns

应用 C: MAX (((5H+2M)+3M)+M, ((4H+2M)+4M)+M) = 4H+7M = 60 + 315 = 375ns

(c) FR-FCFS 利用的是内存引用行为的什么特征? (6 个字 ☺)

参考答案：

行缓冲局部性

(d) 请简要描述可以最大化请求吞吐量的调度策略，请求吞吐量的意思是每单位时间响应的请求数。(十个字左右☺)

参考答案：

行缓冲优先的先来先服务(FR-FCFS)

4.2 应用感知的调度策略

图中的 3 个应用，应用 C 是内存密集程度最低的(即有最少的请求数)。然而，它经历了最长的停顿时间，因为它的请求响应晚于其它多个被优先服务的应用的请求。为了保证应用 C 的停顿时间最短，可以为它的请求分配最高优先级，而给应用 A 和 B 的请求分配同样的低优先级。

(a) 调度策略 X: 当应用 C 分配高优先级并且应用 A 和 B 分配相同的低优先级，每个应用的停顿时间是多少? (对于相同优先级的请求，假设使用 FR-FCFS 策略)

参考答案：
应用 A: MAX (M+(4H+3M), M+(3H+3M)+4M) = 3H+8M = 45+360 = 405ns
应用 B: MAX (M+(4H+3M)+3M, M+(3H+3M)) = 4H+7M = 60+315 = 375ns
应用 C: MAX (M, M) = M = 45ns

你能否设计一个更好的调度策略? 虽然应用 C 的停顿时间小了，但是应用 A 和 B 之间还是会互相影响。

(b) 为其它两个应用分配优先级，这样你可以最小化所有应用的平均停顿时间。请具体从大到小列出三个应用的优先级(对于相同优先级的请求，假设使用 FR-FCFS 策略)

参考答案：
C>B>A

(c) 调度策略 Y: 使用你的新调度策略，每个应用的停顿时间分别是多少? (对于相同优先级的请求，假设使用 FR-FCFS 策略)

参考答案：
应用 A: MAX (M+(3M)+(4H+3M), M+(3H+3M)+4M) = 3H+8M = 45+360 = 405ns
应用 B: MAX (M+(3M), M+(3H+3M)) = 3H+4M = 45+180 = 225ns
应用 C: MAX (M, M) = M = 45ns

(d) 请将四种调度策略 (FCFS, FR-FCFS, X, Y)的平均停顿时间从大到小排列

参考答案：
Y<X<FR-FCFS<FCFS

5 分层存储体系结构 10 分

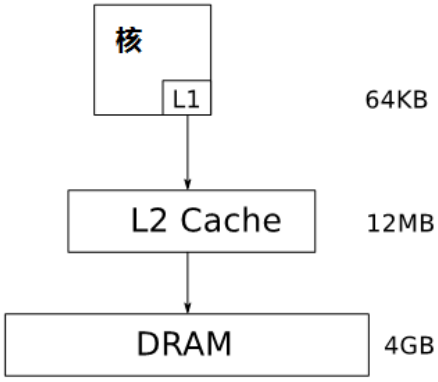
假设你研究出了下一代的存储技术: “魔法 RAM”。魔法 RAM 的位元是非易失性的；它的访问延迟是 SRAM 的 2 倍，与 DRAM 相同；读/写时的能耗和成本与 DRAM 相当；比 DRAM 的密度更高。然而，魔法 RAM 有一个缺点: 每个位元在执行 2000 次写操作之后会停止运转。
(a) 相比 DRAM，魔法 RAM 除了密度高之外，还有什么优势么? 请解释。

参考答案：
是的。
魔法 RAM 不需要刷新，因为它是非易失性的。这可以降低动态功耗，总线的占用和 bank 的竞争。魔法 RAM 的非易失性还可能有利于新的使用模式或编程模型。

(b) 相比 SRAM，魔法 RAM 有什么优势吗? 请解释。

参考答案：
是的。
魔法 RAM 有更高的密度和更低的成本。

(c) 假设一个系统有 64KB SRAM 的 L1 cache 、12MB SRAM 的 L2 cache 和 4GB DRAM 的主存。

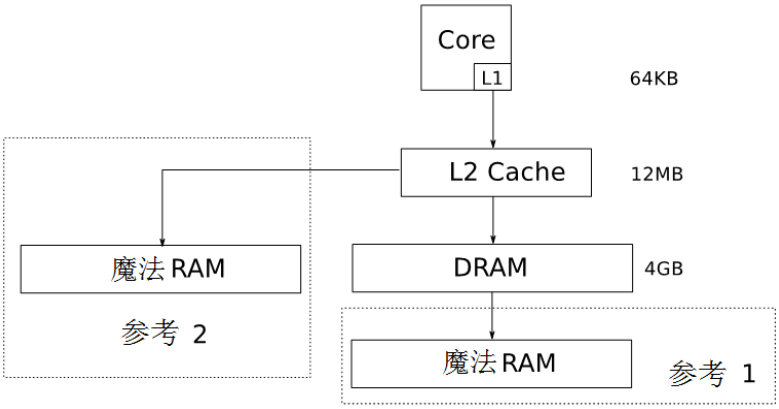


假设你可以利用这个分层存储结构，经过自由地设计和增加任何结构以克服魔法 RAM 的缺陷(除了修改魔法 RAM 本身)
(i) 可能将魔法 RAM 加入这个分层存储结构以减小它的缺陷吗?

参考答案：
是的。

(ii) 如果可能, 魔法 RAM 该放到哪里? 根据上面的图来说明，并说明为什么选择放在这个位置。
如果不可能，为什么? 请解释。

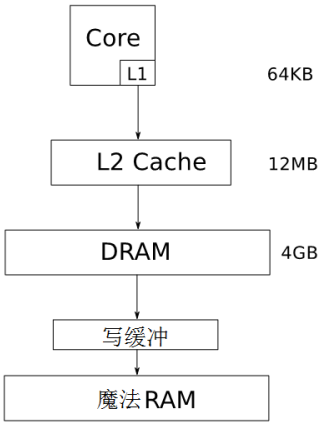
参考答案：
可能的正确答案不止一种。
其中一种: 在存储的层次结构中，将魔法 RAM 放置于 DRAM 之下，利用 DRAM 作为魔法 RAM 的 cache。这样，由 DRAM 执行 更多的写操作，使魔法 RAM 不会过快的磨损。
另一种是把魔法 RAM 与 DRAM 并排放置 (相同或不同的通道上)，利用魔法 RAM 显式地处理只读数据。



(d) 请给出一种通过修改这个分层存储结构以减少或克服魔法 RAM 缺陷的方法。请简单清晰地说明你的方法，可以利用图示来说明问题。

参考答案：

采用上述参考 1 的方案，需要增加一个联合写缓冲的组件以减少写操作对魔法 RAM 的损耗。同时，存储层次结构中也应该提供一些损耗均衡的机制，或者预测哪些数据被修改的可能性低，将这些数据存入魔法 RAM。



6 虚存和 cache 10 分

一个 2 路组相联 cache，采用写回策略和 LRU 替换策略，需要 $15x2^9$ bit 的标签存储来存储包括有效位、脏位以及 LRU 等标签信息。Cache 虚拟索引，物理标签。虚拟地址空间 1MB，页大小是 2 KB，cache 块大小是 8 字节。

(a)该 cache 的数据存储是多少字节？

参考答案：
8 KB

Cache 是 2 路组相联，所以每个组有两个标签，每个标签 t 位，2 位有效位，2 位脏位，1 位 LRU 位。
如果我们用 i 表示索引位的位数，
标签存储大小= 2^i x (2 x t + 2 + 2 + 1) = 15 x 2^9
所以，2t = 10，t = 5
i= 9
数据存储大小 = 2^i x (2 x 8) = 2^9 x (2 x 8) = 8 KB

(b)虚拟索引中有多少位来自虚页号？

参考答案：
1 位
页大小为 2 KB，因此页偏移量是 11 位 (10:0)；
cache 块偏移量是 3 位 (2:0)，虚拟索引 9 位 (11:3)；
所以虚拟索引中有 1 位 (11) 来自虚页号。

(c)这个存储系统的物理地址空间有多大？

参考答案：
64 KB
页偏移量 11 位；物理帧号(物理标签)是 5 位；
所以，物理地址空间是 $2^{(11+5)} = 2^{16}$ = 64 KB.