

CA_Assignment11

 Assign	
 tag	homework
 姓名	周鹏宇
 学号	2019K8009929039

1. 关于多核处理器的 Cache 结构，请介绍 UCA 与 NUCA 的特点。

- UCA 是一种集中式共享结构，多个处理器核通过总线或者交叉开关连接 LLC，所有处理器核对 LLC 的访问延迟相同。这种集中式的共享 LLC，很容易随着处理器核数目的增加成为瓶颈。另外，UCA 结构由于使用总线或者交叉开关互连，可扩展性受限。
- NUCA 是一种分布式共享结构，每个处理器核拥有本地的 LLC，并通过片上互连访问其他处理器核的 LLC。在 NUCA 结构中，处理器核可以访问所有的 LLC，但是不同位置的 LLC 有不同的访问延迟。采用基于目录的 Cache 一致性协议，具有良好的可扩展性，可以有效支持较多数目的处理器核。

2. 有两个并行执行的线程，在顺序一致性和弱一致性下，它各有几种正确的执行顺序，给出执行次序和最后的正确结果(假设 X、Y 的初始值均为 0)

P1	P2
X=1; print Y;	Y=1; print X;

- 顺序一致性：
 - X=1 Y=1 print Y print X 结果为1 1（顺序为X Y，后略）
 - X=1 print Y Y=1 print X 结果为0 1
 - X=1 Y=1 print X print Y 结果为1 1
 - Y=1 X=1 printX print Y 结果为 1 1
 - Y=1 print X X=1 print Y 结果为0 1

- Y=1 X=1 printY print X 结果为 1 1
- 弱一致性：
 - 不妨记print为同步操作，则必然先完成X=1 和 Y=1后，再print X和print Y，共有四种顺序，但结果都是1 1

3. 关于 Cache 一致性协议，MESI 协议比 ESI 协议增加了 M 状态，请解释有什么好处。

增加一个 Modified 状态的优点是减少了 Cache 到内存的数据传输次数，Cache 只需要将 Modified 状态的 Cache 行写回内存。

4. 请分别采用 Fetch_and_Increment 和 Compare_and_Swap 原子指令编写实现自旋锁的代码，并分析可能的性能改进措施。

```
//fetch and increment
typedef spinlock{
    int status;
}spinlock_t;

void aquire_lock(spinlock *lock){
    while (Fetch_and_Increment(lock.status));
}

void release_lock(spinlock *lock){
    lock.status = 0;
}
```

```
//compare and swap
typedef spinlock{
    int status;
}spinlock_t;

void aquire_lock(spinlock *lock){
    int flag = 1;
    while (flag){
        Compare_and_Swap(&lock.status, 0, &flag);
    };
}

void release_lock(spinlock *lock){
    lock.status = 0;
}
```

Test_and_Set 自旋锁最主要的一个缺点就是对锁变量的访存冲突。当一个处理器核获得锁以后，其他等待的处理器核会不断循环执行 Test_and_Set 指令访问锁变

量，试图获取锁权限，从而在片上互连上产生大量的访存通信。一种简单的优化方法就是在 Test_and_Set 指令之间加入一定的延迟，减少等待阶段 Test_and_Set 原子指令自旋执行的次数以减轻访存的压力。

5. 在共享存储的多处理器中，经常会出现假共享现象。假共享是由于两个变量处于同一个Cache 行中引起的，会对性能造成损失。为了尽量减少假共享的发生，程序员在写程序时应该注意什么？

- 避免假共享的主要方式是进行代码检查。潜在的假共享主要出现在线程访问全局或动态分配共享数据结构。
- 可以通过内存填充(padding)的方式予以更正，目的是确保引起假共享的变量在内存中存放的位置相隔足够远，从而不会驻留在同一高速缓存块中。

6. 请介绍片上网络路由器设计中的虚通道概念，并说明采用虚通道有什么好处。

虚通道是用于描述ATM（异步转移模式）中信元进行单向传输的一个概念

虚通道可以分时复用物理通道，节省片上资源

7. 分析 Fermi GPU 的存储结构，指出不同层次存储结构的带宽、延迟、是否共享。

存储结构	位置	容量	带宽	延迟	共享	备注
<u>SM 寄存器堆</u>	SM 内部	128 KB	8000 GB/s	1 cycle	否	
<u>共享存储</u>	SM 内部	16 KB / 48 KB	1000 GB/s	20-30 cycles	是	与 L1 Cache 总计 64 KB 每 SM
<u>L1 Cache</u>	SM 内部	16 KB / 48 KB	1000 GB/s	20-30 cycles	否	与共享存储总计 64 KB 每 SM
<u>L2 Cache</u>	GPU	768 KB	-	-	是	未找到带宽、延迟的数据
<u>主存储</u>	DRAM	最大 6GB	177 GB/s	400-800 cycles	是	GDDR5，与 CPU 共享