

1 Consistency Tutorial

基于共享缓存的内存一致性模型

focus on consistency models proposed for hardware-based shared-memory systems

内存一致性模型的影响

programmability 可编程性、performance 性能、portability 移植性兼容性

缓存一致性 (1) (2) 和顺序一致性 (1) (2) (3) (4), Cache Coherence and Sequential Consistency

- (1) 写操作的结果最终会被所有处理器可见
- (2) 对同一个地址的写操作被所有处理器以相同的顺序看见
- (3) 对所有地址的写操作被所有处理器以相同的顺序看见
- (4) 任意一个处理器均是以程序顺序执行操作

写操作的原子性幻象

存在缓存的情况下，系统需要一个方法来获知缓存是否接受到了无效 (invalidation) 或更新 (update) 信息。只有当通知信息 (无效 or 更新) 全部完成后，处理器才能视一个写操作为完成。需要满足两个条件才能达成写操作的原子性

1. 对同一个地址的写操作被所有处理器以相同的顺序看见
2. 禁止一个读返回最近一个写的结果，直到这个写操作被通知到所有的缓存副本中

顺序一致性，sequential consistency

- 1 程序顺序要求，program order requirement
- 2 写操作原子性要求，write atomicity requirement

宽松的一致性模型

- 1 对程序顺序要求的宽松：WR, WW, RR/RW
- 2 对写操作原子性的宽松：read other's/own write early

一致性的宽松将复杂度推到了编程级别，严格的顺序一致性则将复杂度隐藏在处理器内部

宽松的一致性模型可以得到更好的性能，但是需要在编程级别提供一些信息供底层处理，以免发生与编程意图相违背的结果，编译器需要负责将这些信息交给硬件处理

| Relaxation | W \rightarrow R Order | W \rightarrow W Order | R \rightarrow RW Order | Read Others' Write Early | Read Own Write Early | Safety net |
|-----------------|----------------------------|----------------------------|-----------------------------|-----------------------------|-------------------------|------------------------------|
| SC [16] | | | | | ✓ | |
| IBM 370 [14] | ✓ | | | | | serialization instructions |
| TSO [20] | ✓ | | | | ✓ | RMW |
| PC [13, 12] | ✓ | | | ✓ | ✓ | RMW |
| PSO [20] | ✓ | ✓ | | | ✓ | RMW, STBAR |
| WO [5] | ✓ | ✓ | ✓ | | ✓ | synchronization |
| RCsc [13, 12] | ✓ | ✓ | ✓ | | ✓ | release, acquire, nsync, RMW |
| RCpc [13, 12] | ✓ | ✓ | ✓ | ✓ | ✓ | release, acquire, nsync, RMW |
| Alpha [19] | ✓ | ✓ | ✓ | | ✓ | MB, WMB |
| RMO [21] | ✓ | ✓ | ✓ | | ✓ | various MEMBAR's |
| PowerPC [17, 4] | ✓ | ✓ | ✓ | ✓ | ✓ | SYNC |