

20 监听一致性:

优点: 实现简单, 容易理解, 能够保证数据的实时性, 因为客户端只有在接收到服务器处理完成的确认信息之后才能执行下一步操作。

缺点: 需要频繁地向服务器发送消息进行确认, 会产生较大的网络延迟, 特别是在集群规模较大的场景下, 容易出现网络瓶颈。

目录一致性:

优点: 能够减少客户端和服务器之间的通信, 有效降低网络延迟, 适用于数据读取频率高的场景。

缺点: 实现相对复杂, 需要额外的服务器来维护数据的版本信息。

缓存一致性的实现代价:

硬件成本: 实现缓存一致性需要更多的硬件支持。

性能: 缓存一致性协议需要处理器进行许多额外的操作, 可能会导致性能下降。

能耗: 数据传输和缓存控制等操作需要耗费更多的电能, 可能会导致处理器的功耗增加。

软件: 实现缓存一致性需要相应的软件支持。软件需要按照缓存一致性协议编写代码, 这可能会增加软件开发的复杂性和代价。

19 (1) 因为它被用来区分不同的缓存行, 如果两个或更多的缓存行具有相同的低位标签, 缓存一致性协议将无法判断哪个缓存行应被使用, 从而导致数据的一致性。

(2) 微标签技术会改变缓存行的标识方式, 需要在记录缓存行的最近使用状态同时记录其对应的微标签, 意味着缓存替换策略算法需要考虑更多的细节信息, 同时其可能会影响缓存行的数量和访问方式, 需要考虑缓存组的数量和组索引的大小。

(3) $8KB = 2^{13}$ 字节 $16KB = 2^{14}$ 字节 每个缓存块大小 2^{11} 字节 用11位表示



扫描全能王 创建

$\text{Index} = \text{address} \bmod 4$ 共 6 组，用 6 位表示组索引和低位标签
 $2^6 \text{ byte} = 2^9 \text{ bit}$

17 (1) 块大小 64 字节，块内偏移 6 位， $14 - 6 = 8$ 位虚拟页号：索引 $\text{Index} = 2$ 位，16 个条目 4 组 $16 \bmod 4$ ，剩下 6 位为标签 $0x05a4 = 0000\ 0101\ 1010\ 0100$ 取前 6 位为标签
 $\text{tag} = 00\ 0101 = 0x05$ 在组号 2 中 $\therefore \text{TLB命中 } 0x1C = 01\ 1100$ 块内偏移
与 $0x05a4$ 后 6 位 $10\ 0100$ 物理地址 $01\ 1100\ 10\ 0100 = 0x\ 0724$

(2) 虚拟页号有 8 位， $2^8 = 256$ 个条目

(3) 块大小 4 字节 块内偏移 2 位，物理地址 12 位，块地址 $12 - 2 = 10$ 位 16 组 $\text{Index} = 64 \bmod 16 = 4$ 即 4 组 6 位 tag $= 011100 = 0x1C$ 在第 9 组 命中缓存
块内偏移 $0x00$ $\text{Index} = 1001$ 对应第 9 组 取这一组的 $0x0$ 得到 $0x63$

18 (1) A B C D A B C D 命中率 50%

way0 — A A C C A A C

way1 — — B B D D B B

命中 N N N N N N /V N

(2) 不替换时 命中率 50%，始终预测 A 与 B



扫描全能王 创建