AMAT: average memory access time

AMAT = Hit Time + MissRate × MissPenalty

1. Reduce Hit Time
2. Reduce cache size (miss rate ↑)
3. Reduce cache associativity (miss rate↑)
4. Overlap one hit with other

Pipelined (L1) Caches (multiple cycles to access, dividing access to multiple stages):

* Address index
* Read tags and valid bits of each block in set
* Data read

1. TLB and cache hit (virtual address -> TLB (frame number) -> physical address)

Instead, use virtual address to address data (no TLB latency), but:

1. TLB have permission (need to access for permission check) -> no advantage, still need to access TLB) …… 17.6
2. Virtual address is specific for one process -> need perform cache flush every context switch (expensive) …… 17.6

Combine virtually indexed and physically tagged cache:

* A virtual address: tag + virtual index + cache offset, use virtual index to access data, also, parallel use TLB get tag and check whether hit (no need to worry for context switch, since tag is check) -> no need to flush cache, but still need to flush TLB (TLB flush penalty is small, comparing with cache flush. But write-back policy is time-consuming)

图片包含 文字, 人员

描述已自动生成

* Aliasing problem: (multiple virtual index to same physical address, thus, might same data in different virtual cache