Requested Word First or **Critical Word First**

Cache read miss일 때, 메인 메모리에서 데이터를 불러와 순차적으로 반환하지 않고, 요구한 word를 먼저 보내는 방법.

Write-Through

Cache write hit일 때, cache를 업데이트 하면서 동시에 메인 메모리도 업데이틀 하는 방법. 큰 오버헤드를 발생시키므로 **write buffer**를 사용한다. 하지만 buffer가 꽉차게 되면 결국 CPU는 기다려야 한다.

- Allocate on miss

  블록에 데이터를 올린다.

- Write around

  블록에 데이터를 올리지 않고, 바로 메인메모리에 쓴다. 프로그램을 초기화 할 때 이러한 방법을 사용한다.

Write-Back

Cache write hit일 때, cache만 업데이트하고 **dirty bit**을 표시한다. 업데이트 된 데이터가 cache에서 추방될 때, 메인 메모리에 업데이트 한다. 물론 buffer사용이 가능하다.

Measuring Cache Performance

$$\text{Memory stall cycles}$$

$$= \frac{\text{Memory accesses}}{\text{Program}} \times \text{Miss rate} \times \text{Miss penalty}$$

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Miss}}{\text{Instruction}} \times \text{Miss penalty}$$

Average Memory Access Time (AMAT)

$$\text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

Associative Caches

하나의 cache라인이 여러 개의 블록을 가지는것

- Fully associative

  블록이 cache의 어느곳에나 위치할 수 있으므로 해당 블록을 찾기위해 cache의 전수조사가 필요하다.

- n-way set associatve.

  Cache를 (block/n)개의 set으로 나누고, 블록은 지정된 set의 어느곳에나 위치할 수 있다.

Least Recently Used (LRU)

Associative chache에서, set 안의 블록을 교체할때, 가장 오랫동안 사용되지 않은 블록을 교체하는 방법이다.

Multilevel Cache Considerations

Primary(L-1) cache는 hit time을 줄이고, L-2 chache는 miss rate를 줄이는데에 초점을 둔다. 결과적으로 L-1 cache는 크기를 작게 하고 miss panalty를 줄이기 위해 블록크기가 작다. L-2 cache는 반대이며, miss rate를 줄여야 하므로 높은 associative를 가진다.

Translation Lookaside Buffer (TLB)

최근에 address mapping된 정보를 가지고 있는 작은 buffer.

Compulsory misses (aka cold start misses)

블록을 처음 접근 할 때 발생하는 miss

Capacity misses

상위레벨 cache가 하위레벨 cache보다 작기 때문에 발생하는 miss

Conflict misses (aka collision misses)

Non-fully associative cache에서 하나의 set의 entry가 경쟁하면서 발생하는 miss

Invalidating Snooping Protocols

Cache가 bus를 모니터링 하다가, 다른 cache에서 write이 발생하면 자신의 cache를 비운다. 그리고, 자신에게 read가 발생하면 메인 메모리에서 읽어온다.

Mean Time to Failure (MTTF)

Reliability: failure발생 후 다음 failure가 발생하기 까지의 평균 시간.

Mean Time to Repair (MTTR)

Service interruption: failure을 복구하는 평균 시간.

Mean Time Between Failures (MTBF)

MTBF = MTTF + MTTR

Availability

Availability = MTTF / (MTTF+MTTR)
= MTTF / MTBF

실제 어떤 서비스를 받을 확률. 클수록 좋다.

Improving Availability

Increase MTTF, Reduce MTTR.

Disk Sectors and Access: Sector

각 섹터는 ID를 가지고 있으며, Error Correcting Code(ECC)를 포함한 512byte로 이루어져 있다. 섹터와 섹터를 구분해야하므로 섹터와 섹터사이에 약간의 갭이 존재한다.

Access to a sector involves

1. Queuing delay

   접근하고자 하는 섹터에 접근하기 전에, 이전의 요청을 완료하는데 걸리는 시간.

2. Seek

   섹터를 찾는 과정. 걸리는 시간은 Seek time이라고 한다.

3. Rotational latency

   섹터는 플랫터에 존재하고, 읽기위해 head 아래에 위치하여야한다. 따라서 플랫터를 회전시켜 head 아래에 위치시키는 시간이 존재한다.

4. Data transfer

   섹터의 데이터를 읽거나 쓰는데 걸리는 시간이 존재한다.

5. Controller overhead

   Controller를 통해 읽은 데이터를 I/O 버스에 전달하는데 걸리는 시간.

Average read time

= Average seek time + Rotational latency + Transfer time + Controller delay

Average read time Example

512B sector, 15,000 rpm, 4ms average seek time, 100MB/s transfer rate, 0.2mss controller overhead, idle disk

Queuing delay = zero (idle disk)

Average seek time = 4ms

Average rotational latency = (1/2)/(15,000/3600) = 2ms (unit convert)

Transfer time = 512B/(100MB/s) = 0.005 ms

Controller overhead = 0.2ms

all add = 6.2ms

Polling and interrupt-driven I/O

CPU가 개입하여 메모리와 I/O데이터 레지스터 사이에서 데이터를 전송한다.

Direct Memory Access (DMA)

OS가 프로그램 메모리 시작주소를 알려주고 I/O controller가 알아서 데이터를 주고 받는 것. 전송이 모두 끝나면 CPU에 interrupt를 발생시킨다. 즉 데이터 전송에 CPU의 개입이 없다. Throughput을 향상시킨다.

DMA: Cache Interation

Cache가 DMA를 위해 사용되게 되면 해당 블록을 flush 시킨다. 또는 I/O를 위해 non-cacheable memory location을 지정한다.

Redundant Array of Inexpensive Diskts (RAID)

여러 개의 작은 disk로 구성되어 있으며, 병렬성으로 성능을 향상시킨다.

- RAID 0

  No redundancy. 데이터를 여러 개의 disk에 stripping한다. 선능은 2배 빨라지지만, 데이터에 오류가 생기면 복구할 방법이 없다.

- RAID 1: Mirroing

  N+N disks. 데이터를 중복해서 쓰므로 성능향상은 없다. 하지마 똑 같은 데이터가 존재하므로 오류로부터 복구가 가능하다.

- RAID 2: Error Correcting Code (ECC)

  N+E disks. E는 ECC를 위한 disk이다. bit-level로 N디스크에 나누어 저장되고, ECC를 생성하여 E디스크에 저장된다. 너무 복잡하므로 사용되지 않는다.

- RAID 3: Bit-Interleaved Parity

  N+1 disks. Byte-level로 N disk에 stripping된다. 남은 1개의 disk에 parity를 저장한다. 읽을 때 N disk를 읽고, 쓸 때 N disk를 읽어 parity를 생성한다. 오류가 발생 할 때 parity를 이용하여 복구한다. 거의 쓰이지 않는다.

- RAID 4: Block-Interleaved Parity

  N+1 disks. Block-level로 N disk에 stripping되는 것만 다르고 RAID3와 유사하다. 읽을 때, 필요한 블록이 존재하는 disk만 읽는다. 쓸 때, 수정되어야 할 블록과 parity disk를 읽어 업데이트한다. 거의 사용되지 않는다.

- RAID 5: Distributed Pariry

  N+1 disks. RAID 4와 같지만, parity 블록은 stripping한다. Parity disk의 병목현상을 완화시키므로 parity disk에서 오류가 발생할 확률이 감소한다. 많이 사용된다.

- RAID 6: P+Q Redundancy

  N+2 disks. RAID 5와 같지만, 2개의 parity disk를 사용한다.

Principle of Locality

프로그램에서 특정한 시간에 메모리 주소에서 아주 적은부분만을 접근한다.

- Temporal locality

  최근에 접근한 메모리를 가까운 시간에 다시 접근할 확률이 높다.

- Spatial locality

  최근에 접근한 메모리 근처의 메모리는 가까운 시간에 접근될 확률이 높다.

DRAM: Burst mode

같은 열의 데이터를 조회할 경우, 행값은 고정시키고 열값만 바꾸어 latency를 줄인다.

Double Data Rate (DDR)

Rising과 falling clock edge 둘 다에서 데이터 전송이 가능하다.

Quead Data Rate (QDR or DDR2)

DRAM의 I/O포트가 각각 두개씩 있는것으로, 대역폭이 2배 증가한다.

Example cache block read

1 bus cycle for address transfer

15 bus cycles per DRAM access

1 bus cycle per data transfer

For 4-word block, 1-word-wide DRAM

Miss penalty = 1+4*15+4*1 = 65 bus cycles

Bandwidth = 16bytes / 65cycles = 0.25B/cycle

4-word wide memory

Miss penalty = 1+15+1=17 bus cycles

Bandwidth = 16bytes / 17cycles = 0.94B/cycle

4-bank interleaved memory

Mis penalty = 1+15+4*1 = 20 bus cycles

Bandwidth = 16bytes/20cycles = 0.8B/cycle