Universiteit van Amsterdam

Samenvatting

Hoofdstuk 5, Exploiting memory hierarchy

6 oktober 2014

Student:
Steven Raaijmakers
10804242

Tutor:
Ellen van Leeuwen

PAV-groep:
COBOL (C1)

1 5.1 Introduction

Bij het lezen en schrijven van data wil je snel toegang hebben tot de data. Er zijn twee principes van lokaliteit:

- Temporal locality: een gebruikt item moet snel weer toegankelijk zijn
- Spatial locality: een gebruikt item moet dichtbij andere soortegelijke items staan

De memory hierarchy is een hierarchie waarin de snelste en duurste memory bovenaan staat. Hoe lager in de hierarchie; hoe trager en goedkoper de memory. Een hit is het moment waarop de gezochte data gevonden wordt. Als de data niet gevonden wordt, noemt men dit een miss.

2 5.2 Memory technologies

De memory hierarchy maakt gebruik van vier hoofdzakelijke technologin:

- SRAM: memory met een enkele toegangspoort die gelezen of geschreven kan worden. Het voordeel van SRAM is dat het niet hoeft te refreshen. Het behoudt de data zolang er energie verkregen wordt. Een SRAM maakt gebruik van 6 transistors per bit.
- DRAM: maakt gebruik van slechts 1 transistor per bit waardoor het veel goedkoper is, en relatief meer data kan opslaan. Aan de andere kant verbuikt DRAM meer stroom.
- Flash memory: is memory die elektrisch verwijdert en geherprogrammerd kan worden. Wear leveling

- Disk memory: de toegang tot data op een Disk memory gaat als volgt; seek, hierbij wordt de head op de juiste track geplaatst. Vervolgens moet de track draaien zodat de juiste sector onder de head wordt geplaatst (rotational latency). Tot slot wordt de data verstuurt in een blok van bits (transfer time).

3 5.3 The basics of caches

Cache is de plek waar data opgeslagen wordt zodat het toegankelijk is. De cache is geplaatst op de processor en is hierdoor erg snel. Een adresverwijzing vanuit de cache wordt verdeeld in: een tag (deze tag wordt vergeleken met mogelijke tags in de cache) en een cache index (wordt gebruikt om het juiste block te selecteren). Als er een cache miss optreedt wordt het originale PC naar de memory gestuurd, hierna wordt de main memory verteld om te lezen en de te wachten op de memory. Vervolgens wordt de cache beschreven met deze data. Als nu de instructie herhaald wordt zal deze nu wel te vinden zijn in de cache.

4 5.4 Measuring and improving cache performance

De cache kan op verschillende manieren verbeterd worden. Door de miss ratio zo laag mogelijk te laten worden, de miss penalty die optreedt bij een miss te verminderen en tot slot de tijd die nodig is voor een hit verlagen. De miss-ratio kan verlaagd worden door meer en grotere caches te gebruiken. De miss penalty kan aanzienlijk minder worden door bijvoorbeeld de memory in verschillende vakken op te delen waarin tegelijkertijd gezocht kan worden.

5 5.7 Virtual memory

Virtuele memory is wanneer de main memory zich als een cache gedraagt. Virtuele memory zet het adres van een programma om in een fysiek adres. Wanneer de juiste bit voor een virtuele pagina uitstaat zal er een page fault voorkomen. Deze fout wordt afgehandelt door het OS, die vervolgens de pagina moet vinden in een hogere hiararchie en deze in de juiste memory plaatst. Processoren hebben een cache die recent gebruikte translaties opslaat (translation-lookaside buffer).

6 5.8 A common framework for memory hierarchy

Een block kan op verschillende plekken geplaatst worden. Zie pagina 455 voor de voorwaardes. Een block kan gelocaliseerd worden op basis van een van te voren afgesproken block-placement-scheme. Als er een cache miss optreedt moet er een block vervangen worden, dit kan een random block zijn of het laatst gebruikte block (least recentley used). Een key-onderdeel van een memory hierarchy is hoe het omgaat met writes. Twee basis opties zijn de write-through, waarbij de informatie in een block in het cache geschreven wordt en tevens in een block van een lager level, en write-back waarbij de informatie slechts in een block in de cache geschreven wordt. Het bewerkte block wordt geschreven naar een lager level maar enkel als dit vervangen wordt. De drie C's: Compulsory misses (cache misses veroorzaakt doordat een block nooit in de cache geweest is). Capacity misses (cache misses veroorzaakt doordat de cache de groote van het aantal blocks niet aankan). Conflict misses (cache misses verooraakt doordat dezelfde blocken worden gebruikt voor een enkele set).

7 5.9 Using a FSM to control a simple cache

Een finite-state machine bestaat uit een set van states en directions van hoe deze states verandert kunnen worden.

8 5.15 Fallacies and pitfalls

Belangrijke valkuilen bij de memory hierachy in computer architectuur zijn de volgende. Vaak wordt bij het schrijven van programma's geen rekening gehouden met het memory system, terwijl hiermee de prestaties van een programma zouden kunnen worden verdubbeld. Hiernaast houdt men vaak ook geen rekening met de byte adressing bij de simulatie van een cache. Een processor moet zoveel mogelijk instructies blijven uitvoeren. Wanneer er een mis optreed is het dus belangrijk dat er geen stalls ontstaan maar dat er alvast een andere instructie wordt uitgevoerd.

9 5.16 Concluding remarks

Het principe van localiteit geeft het meeste tijdwinst bij het ophalen van data. De localiteit kan verbeterd worden door de software maar ook door de hardware. Ook kan er gebruik worden gemaakt van prefetching waarbij een

blok data al in de cache gezet wordt voordat hierom gevraagd wordt. Dit is een vorm van predict.