Universiteit van Amsterdam

Samenvatting

Hoofdstuk 6: Parallel Processors from Client to Cloud

13 oktober 2014

Student:
Steven Raaijmakers
10804242

Tutor:
Ellen van Leeuwen

PAV-groep:
COBOL (C1)

6.1 Introduction

Een multiprocessor is een computer met tenminste twee processoren. De software die hierop draait moet zo worden geschreven dat het gebruik kan maken van de verschillende processoren.

Meer througput voor de taken wordt task-level parallelism genoemd. Het houdt in dat n taak dus door meerdere processoren wordt uitgevoerd (parallel processing program).

6.2 The Difficulty of Creating Parallel Processing Programs

Het schrijven van software voor een parallel architectuur is moeilijk door snelheden die je ermee kan bereiken. Zo verwachten men gewoonlijk dat 100 processoren tegenover 1, een versnelling van 100 zal geven. De wet van Amdahl schrijft voor dat dit echter niet zo is:

Speed-up = 1 / (1- Fraction time affected) + (Fraction time affected / Amount of improvement)

Strong scaling: speed-up wordt verkregen zonder het probleem te vergroten Weak scaling: speed-up wordt verkregen maar het probleem wordt vergroot door het toenemende aantal processoren.

6.3 SISD, MIMD, SIMD, SPMD and Vector

Een uniprocessor met een enkele stream en single data stream noemt men een SISD. Een multiprocessor met meerdere instructie streams en multiple data streams wordt een MIMD genoemd.

Een SPMD (single program multiple data) is een programma waarbij meerdere processoren elk een andere taak krijgen voor een en hetzelfde programma.

Bij een SIMD (single instruction stream multiple data streams) wordt een en dezelfde instructie aan verschillende data streams toegewezen, zoals bij een vector processor. Voor parallelisme in SMID moet de data dus telkens identiek gestructureerd worden, dit noemt men data level parallelisme.

Vector

Bij een vectorarchitectuur worden data elementen van de memory opgehaald, in grote registers geplaatst, en achtereenvolgens bewerkt volgens de pipeline units. Tot slot wordt het resultaat van de bewerkte registers teruggeschreven naar de memory.

Vector instructies vergen veel werk, omdat er telkens een gehele loop moet worden uitgevoerd. Wel is elke vector instructie onafhankelijk van de andere instructies, waardoor er geen data hazards ontstaan binnen de instructie. Ook zijn er geen control hazards meer omdat een vector instructie een gehele loop vervangt.

6.4 Hardware Multithreading

Hardware multithreading (HM) maakt gebruik van meerdere threads de dezelfde functional units van een enkele processor delen. Hiervoor moet de processor dus wel elke state dupliceren.

Fine-grained multithreading schakelt tussen threads voor elke instructie waardoor een uitvoering meerdere threads beslaat.

Coarse-grained multithreading schakelt alleen tussen threads wanneer er grote stalls optreden, zoals bij bijvoorbeeld last-level cache misses.

Simultaneous multithreading is een variant van HM waarbij meerdere threads gebruikt worden en parallelisme en pipeline gecombineerd worden.

6.5 Multicore and Other Shared Memory Multiprocessors

Een single adress space multiprocessoren worden gebruikt om oude programma goed te laten runnen via parallelisme. Hierbij wordt nog steeds

gebruik gemaakt van meerdere processoren maar slechts een gezamelijk adres, zodat de (oude) programma's niet meer gericht hoefte te worden op waar zich welke data bevond.

Men spreekt van uniform memory access wanneer een word uit de memory onafhankelijk van welke processor hierom vraagt reageert.

Sommige memory kan sneller gealloceerd worden dan andere, afhankelijk van de processor die hierom vraagt. Dit heet nonuniform memory access.

Synchronization is erg belangrijk bij parallelisme, omdat de bewerkte data die parallel wordt uitgevoerd nodig kan zijn bij een andere uitvoering.

OpenMP is een API voor shared memory multiprocessing, geschreven in C.

6.6 Introduction to GPU

Graphics processing is erg verbeterd door de grote vraag naar gaming consoles. Een Graphics Processing Unit (GPU) is een processor die enkel de grafische aspecten hiervan regelt. Er is hiervoor een afwijkende architectuur voor ontwikkeld. CUDA (van NVIDIA) is een taal waarbij programmeurs programma's kunnen laten uitvoeren op GPU.

6.7 Clusters etc.

Een alternatief voor een enkel gezamelijk adres voor meerdere processor is wanneer elke processor zijn eigen adres heeft. Deze multiprocessor moet communiceren over deze locaties op een eigen network. Veel supercomputers bezitten een dergelijk alternatief, het netwerk hiervoor is echter veel duurder dan een lokaal netwerk.

Een cluster bestaat uit meerdere computers die via de I/O verbonden zijn met elkaar waardoor er een message-passing multiprocessor ontstaat. De goedkope kosten, veel voorkomendheid en uitbreidingsmogelijkheden maken een cluster aantrekkelijk voor Internet bedrijven, zoals Google en Facebook.

Warehouse-Scale Computers zijn computers die bestaan uit clusters, op school van 100 duizenden servers. Een populair framework voor WSC is MapReduce. Vlakken waarop je een WSC kan onderscheiden is onder andere dat het schrijven van parallelisme hiervoor makkelijk is. Hiernaast is het zoals eerder gezegd is het ook goedkoper, vooral omdat de WSC gemaakt worden om langer mee te gaan.

6.8 Introduction to MNT

Een network opzetten van multicore chips heeft voordelen en nadelen. Zo kan het duur zijn, gebasseerd op de hoeveelheid switches die bijvoorbeeld

nodig zijn.

6.10 Multiprocessor Benchmarks

Er zijn verschillende benchmarks op de markt die het parallelisme van je computer kunnen meten. In afbeelding 6.16 (bladzijde 541) staat een overzicht van verschillende benchmarks en hun eigenschappen.

De ratio van floating-point operations /byte / memory wordt de arithmetic intensity genoemd.

6.13 Fallacies and Pitfalls

Bij een parallelisme zijn er veel valkuilen. Zo wordt gedacht dat Amdahl's law niet kan worden toegepast op parallel computers. Dit is echter niet zo.

Ook is het belangrijk om moderne software te ontwerpen die op een parallel computer kan draaien. Hiernaast is het ook zeer balngrijk om rekening te houden met de memory bandwith

6.14 Concluding Remarks

Software as a Service wordt steeds belangrijker. Hiervoor levert het gebruik van clusters de beste prestaties. De WSS worden ook steeds belangrijker en veranderen de principes van een server en zijn architectuur.

Tot slot wordt het gebruik van multiprocessoren gezien als een nieuwe verandering die bijbeent met de wet van Amdahl. Het gaat dus niet meer zozeer om het verbeteren van een enkele processor maar om het gebruik van meerdere processoren die gezamelijk een betere prestatie leveren.