

De bedenkers van C++ en Java hebben de afgelopen tijd in de media opgeroepen om te komen tot nieuwe softwareparadigma's. Zowel Bjarne Stroustrup als James Gosling spraken hun zorg uit over de programmeerbaarheid van de nieuwe multi-core processoren. Ondanks de hulp die processorfabrikanten bieden, blijken de grootste inspanningen toch bij de software-developers zelf vandaan te moeten komen.

Multi-core processoren verplaatsen probleem

Softwareontwikkelaars moeten zelf performance uit parallelle hardware halen

Vijf jaar geleden droomden de processorfabrikanten nog van kloksnelheden boven de 10 GHz. Regelmatig werden nieuwe records gebroken. Gebruikers konden voor dezelfde prijs steeds weer een nieuwe generatie processoren kopen die steevast tientallen procenten sneller was dan de vorige. Softwareontwikkelaars op hun beurt hoefden het niet zo nauw te nemen met de efficiëntie van hun code. Het was immers goedkoper om er meer hardware tegenaan te gooien, dan een hoop tijd, geld en energie te investeren in het optimaliseren van hun algoritmen. Als meer hardware even niet kon, dan was het gewoon een kwestie van tijd, wachten tot betaalbare processoren wel genoeg verwerkingskracht hadden. Java is daar een goed voorbeeld van. Inmiddels ziet de wereld er heel anders uit. De snelste X86-processoren die we ooit voor onze desktop-machines hebben kunnen kopen, waren de Pentiums op bijna vier Gigahertz. De laatste generaties Core 2- en Phenom-processoren echter van respectievelijk Intel en AMD worden ongeveer een Gigahertz langzamer geklokt.

Performance per Watt

Voor die toch onverwachte daling van de kloksnelheden zijn diverse oorzaken aan te wijzen. De eerste is de steeds verdere miniaturisering van de chipproductie. Hoe kleiner de afmetin-

gen van de transistoren en de verbindingen op een chip worden, des te inefficiënter ook. Het statische verbruik (de lekkage als de transistoren niet schakelen) van de moderne processorchip is enorm. Daartegenover staat een dynamisch verbruik dat juist afneemt, omdat kleinere chip-technologie met een lager voltage werkt. Om de kloksnelheid te verhogen, zou die spanning juist weer opgevoerd moeten worden, met een sterke (kwadratische) toename van het energieverbruik tot gevolg. Een dergelijke opbouw van het processor-productportfolio past echter niet meer in deze tijd. De aandacht voor duurzaamheid en energieconsumptie, en de gestegen energieprijzen maken dat processorfabrikanten hun producten anders hebben moeten optimaliseren. Waar voorheen de prestatie per prijs de belangrijkste graadmeter was, is dat tegenwoordig de prestatie per Watt. Tekenend is bijvoorbeeld dat de marketinguitingen bij de introductie van Intel's eerste 45 NM-chips volledig in het teken stonden van een hogere performance bij hetzelfde stroomverbruik, en de loodvrije fabricage van de processoren.

Peperduur

Een tweede belangrijke factor heeft te maken met investeringen. Behalve de natuurkundige grenzen die de fabrikanten continu proberen te verleggen,

Adriaan Offerman

X86 multi-core

De huidige multi-core processoren van Intel en AMD bevatten maximaal vier kernen. Daarbij loopt AMD voorop in de integratie daarvan. Hun K10-processoren (de Phenom X4 en de Opteron Barcelona) bevatten daadwerkelijk vier kernen op dezelfde chip. Intel verkoopt ook quad-core processoren (de Core 2 Quad), maar die bestaan uit twee dual-core chips die in één behuizing zijn ondergebracht (Dual-Chip Module, DCM). Ze zijn op het niveau van de Front-Side Bus met elkaar verbonden, gebruikmakend van SMP-logica (Symmetrical Multi-Processing).

Door twee quad-core processoren in een dual-processorconfiguratie te zetten, komt het totaal aantal kernen op acht. Intel en AMD hebben beide deze gelegenheid aangegrepen om op die manier een high-end systeem aan de gaming-wereld te presenteren. Het Skulltrail-platform van Intel werd eerder dit jaar al gelanceerd. Of het FASN8-platform (spreek uit: fascinate) van AMD er nog gaat komen is onduidelijk. Volgens een bericht van The Inquirer zou de ontwikkeling ervan zijn stopgezet. Behalve dat het om heel dure systemen gaat, is het ook nog maar de vraag of gamers er inderdaad iets aan hebben. Hun favoriete shoot-'m-up's moeten een dergelijke grote configuratie immers eerst ondersteunen, en wie echt behoefte heeft aan een dubbele quad-core is misschien beter af met een werkstation.

32 kernen

Zowel Intel als AMD werken op dit moment aan de verdere uitbreiding van het aantal kernen in hun

processoren. Intel's Nehalem-architectuur die over een jaar uit komt, bevat waarschijnlijk acht kernen. Bovendien kan daar nog eens een Graphics Processing Unit (GPU) in dezelfde behuizing bijgezet worden. De opvolger van deze chip heet Sandy Bridge en verschijnt weer twee jaar later. Deze bevat misschien wel 32 kernen. Tot die tijd moeten we het doen met de Dunnington die later dit jaar verschijnt. Deze bevat zes kernen, opgebouwd door drie dual-core processoren tegen elkaar aan te plakken in een Multi-Chip Module (MCM). Met de Itanium-processor zullen vergelijkbare dingen gebeuren. De Tukwilli, de opvolger van de huidige 9100-serie, zal vier kernen bevatten. De opvolger daarvan, de Poulson, zal dit aantal waarschijnlijk nog eens verdubbelen.

Heterogene multi-cores

De roadmap van AMD is een combinatie van multi-core enerzijds en het samenbrengen van general purpose rekeneenheden en GPU's anderzijds. Dit Fusion-project moet resulteren in heterogene multi-cores: chips die meerdere, verschillende kernen bevatten. Later kunnen daar dan bijvoorbeeld ook speciale kernen voor de versleuteling van data bij komen. Hoewel Intel daar minder over bekend heeft gemaakt, is dat wel een richting waar ook die fabrikant aan werkt. De eerste AMD Fusion-producten zullen kleine processoren zijn, gericht op mobiele toepassingen. Over twee jaar verschijnt dan de Sandtiger die is voorzien van acht of zestien kernen.

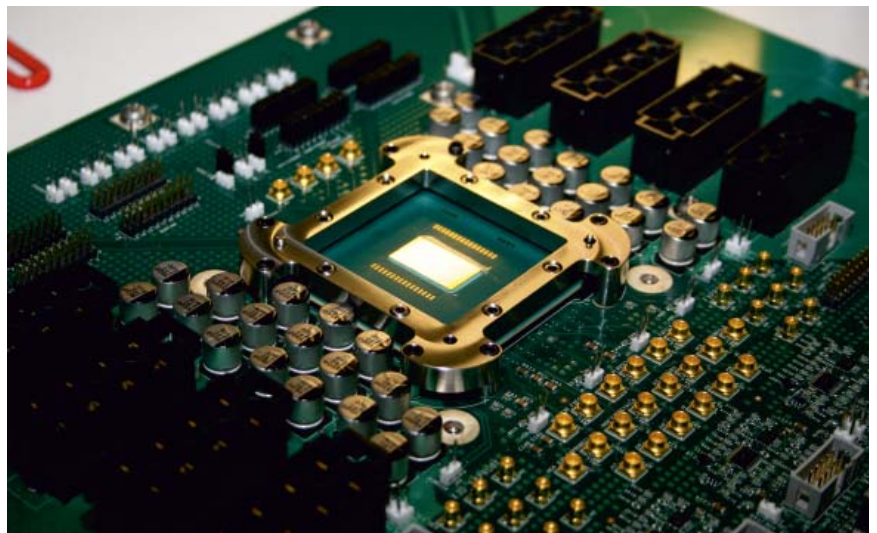
spelen tegenwoordig ook economische grenzen een cruciale rol. Het ontwikkelen en fabriceren van processoren is peperduur. Om een idee te geven: het ontwikkelen van een nieuwe architectuur voor een volgende generatie processoren kost vele honderden miljoenen dollars. Om een volgende stap in het fabricageproces mogelijk te maken, moet een paar miljard uitgetrokken worden. Ben je eenmaal zo ver dat je daadwerkelijk grootschalig wilt produceren, dan gaat het om bedragen die richting de tien miljard dollar gaan. De nieuwe 45 NM-fabriek van Intel in Arizona kostte drie miljard dollar. Dit jaar komen daar nog eens twee fabrieken bij. Het almaar omhoog brengen van de kloksnelheden heeft grote impact op al deze kosten. Kan een fabrikant het aantal Gigahertzen beperkt houden, dan levert dat direct een hoop geld en een snellere doorloop van de ontwikkeling op.

Wet van Moore

Hoewel een lagere kloksnelheid voor de processorfabrikanten zelf heel interessant is, moeten die langzamer geklokte producten natuurlijk wel te verkopen zijn. Voor de meeste gebrui-

kers geldt immers nog steeds: hoe sneller de klok, des te beter de prestaties. Dat mag ook niet verwonderlijk zijn nadat Intel en AMD zelf meer dan vijftien jaar lang de concurrentiestrijd voornamelijk op dat vlak hebben gevoerd. Juist de marketingaspecten hebben ervoor gezorgd dat de fabrikanten deze trend van almaar hogere

Intel Teraflops Motherboard



Een waterval van threads

Van alle fabrikanten van general purpose processoren is Sun het verst als het om multi-core architecture gaat. Deze leverancier heeft zijn UltraSparc-architectuur in twee verschillende lijnen gesplitst. De processor die nu nog de codenaam Rock draagt, wordt de opvolger van de UltraSparc IV. Deze zal 16 kernen bevatten die elk twee threads kunnen verwerken, goed voor 32 threads in totaal. De Rock-processor is vooral geschikt voor grote backoffice-systemen (verticaal schaalbaar) en High-Performance Computing (HPC). Denk daarbij bijvoorbeeld aan grote databases, datamining en wetenschappelijke toepassingen.

Horizontaal schaalbaar

De UltraSparc T1 (codenaam Niagara) bevat maximaal acht kernen die elk vier verschillende threads kunnen draaien. Dat maakt 32 threads in totaal. Waar de HyperThreading-technologie (HTT) van Intel de executionfase van instructies van verschillende threads simpelweg uitvoert wanneer daar gelegenheid toe is (simultaneous multi-threading, SMT), doet de UltraSparc dat anders. De kernen van deze processor schakelen elke

klokcyclus naar een volgende thread (barrel processor). Moet een thread wachten op geheugen of I/O, dan wordt deze tijdelijk uit de pool gehaald. De Niagara-processor is vooral gericht op front-end systemen en applicatieservers (horizontaal schaalbaar). De afzonderlijke kernen van de T1 bevatten dan ook geen floating point-ondersteuning. Met name general purpose code zoals gebruikt door webservers, messaging servers, administratieve toepassingen en Java-applicaties draait lekker op deze systemen.

256 threads

Eind vorig jaar introduceerde Sun de T2 (codenaam Niagara 2). Deze verdubbelt het aantal threads per kern van vier naar acht. Daarmee kan deze processor maximaal 64 threads tegelijkertijd verwerken. Bovendien heeft elke kern nu een eigen Floating-Point Unit (FPU) gekregen. Op dit moment wordt gewerkt aan een multiprocessor-versie van de T2. De chip met de codenaam Victoria Falls zal in SMP-configuraties (Symmetrical Multi-Processing) tot vier processoren kunnen worden ingezet. Dat brengt het totaal aantal threads op 256.

Scott McNealy met de Sun UltraSparc



klokksnelheden überhaupt nog zo lang hebben volgehouden. Een niet onaanzienlijk deel van de prestaties van de moderne processor hebben

we echter niet te danken aan de hoge kloksnelheid, maar aan het steeds grotere aantal transistoren dat op een chip past. De Wet van Moore voorspelt al veertig jaar lang een exponentiële groei: het aantal transistoren per chip verdubbelt elke twee jaar. Hardwarearchitecten vertalen dit grotere aantal transistoren vervolgens naar technische features die de prestaties van de processor moeten verbeteren. Dat kan gaan om relatief eenvoudige verbeteringen zoals grotere caches. Maar ook om heel complexe zaken. Denk dan bijvoorbeeld aan meer Instruction-Level Parallelism (ILP, bijvoorbeeld door middel van pipelining en scoreboarding zoals dat in een superscalar gebeurt), meer speculatieve executie en uitbreidingen van de instructieset (bijvoorbeeld SSE). Daarbij is het goed op te merken dat de hoge kloksnelheden vooral mogelijk zijn gemaakt door de grote caches. Waren die er niet geweest, dan stond de processor nog veel vaker op het relatief langzame geheugen te wachten, en dus niets te doen.

Parallele performance

Net als de zojuist genoemde voorbeelden, is ook het plaatsen van meerdere kernen op dezelfde chip een manier om extra transistoren om te zetten in hogere prestaties. Vervelend is alleen dat softwareontwikkelaars de extra prestaties die nu beschikbaar komen in de vorm van paralleliteit expliciet moeten aanspreken. Ging het om grotere caches, meer ILP, meer speculatie of uitbreidingen van de instructieset, dan bleef de

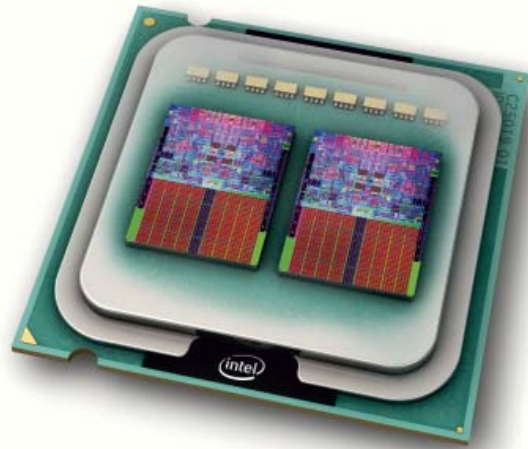
Tera-scale

Intel's Tera-scale onderzoeksprogramma richt zich op many-core processoren voor de desktop, met vele tientallen, later honderden kernen. Behalve om de chips zelf gaat het ook om de programmering daarvan. Al die parallelle hardware heeft immers alleen maar nut als deze ook gebruikt kan worden door de applicaties. Intel heeft inmiddels al de eerste demonstraties van zijn Teraflops onderzoeksprocessor (Polaris) gegeven. De chip bevat maar liefst tachtig kernen, elk bestaande uit (niet veel meer dan) twee Floating-Point Units (FPU's). Op een snelheid van 3,16 GHz brak de processor door de één Teraflops-grens heen, met een stroomverbruik van maar 62 Watt. Omdat de verschillende kernen wel goed met hun burens kunnen communiceren, maar veel minder snel met de buitenwereld (qua architectuur vergelijkbaar met de oude transputers), wil Intel de Teraflops-chip combineren met een geheugenchip door deze twee op elkaar te lijmen. Daarbij worden de kernen met hun stukje geheugen verbonden via speciale verbindingen tussen de twee chips (Thru-Silicon Vias, TSV's).

Throughput computing

Andere projecten onder de Tera-scale-vlag richten zich op de ontwikkeling van tools om de many-cores te kunnen programmeren. Het interessantst is de programmeertaal Ct (C/C++ for Throughput Computing) die in het vorige nummer van Software Release Magazine is besproken. Deze biedt de programmeur allerlei nieuwe datastructuren (data parallel) en flow-statements. Die moeten het gemakkelijker maken code te schrijven die door de compiler goed op een many-core systeem te mappen is. Bovendien moet dezelfde programmatuur ook goed opschalen naar toekomstige processoren met (veel) meer kernen (forward scaling), eventueel voorzien van andere multimedia instructiesets. Daarvoor wordt gebruik gemaakt van intermediaire code en een runtime engine zoals die bij Java ook worden toegepast.

impact daarvan in het slechtste geval beperkt tot de compiler back-end. Gaan we uit van een rechttoe rechtaan single-threaded programma, dan blijven de snelheidsverbeteringen op een multi-core beperkt tot wat in principe ook al met een dual- of quad-processor-systeem in een SMP-configuratie (Symmetrical Multi-Processing) mogelijk zou zijn. In het beste geval heeft het programma een processor(kern) voor zichzelf en kan het ongestoord worden uitgevoerd. De cruciale vraag bij het gebruiken van de nieuwe multi- en straks many-cores is dan ook of de software geschikt (te maken) is voor parallelle verwerking. Voor servertoepassingen zal dat waarschijnlijk allemaal wel goed komen. Netwerkkintensieve applicaties als webservers, mailservers en applicatieserves zijn heel goed uitbreidbaar over SMP-systemen en clusters.



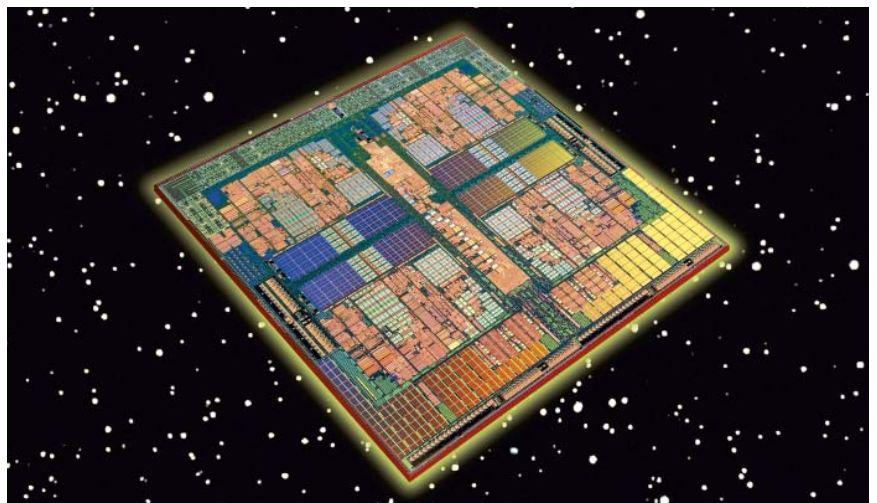
Intel Core 2 Extreme Quad Core

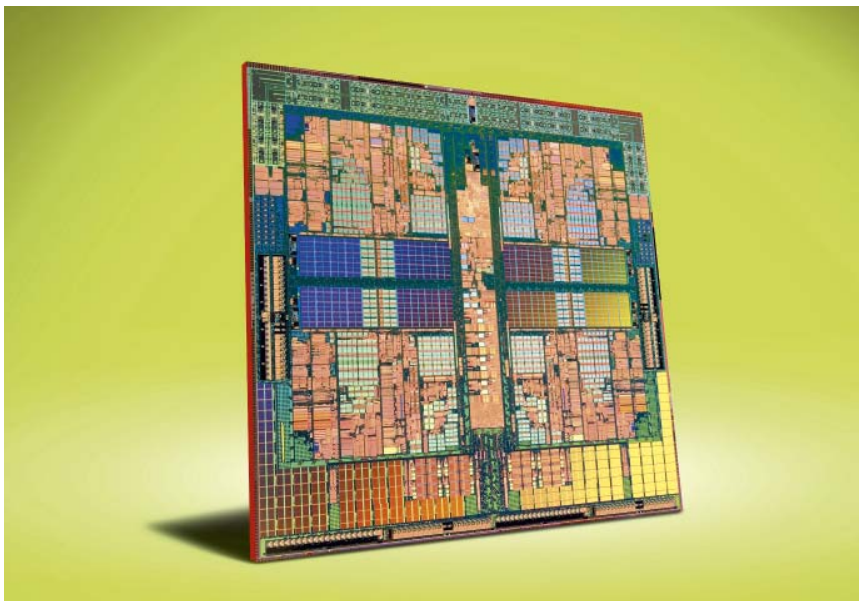
Veel van die toepassingen zijn nu al multi-threaded of multi-process. Men spreekt in dit geval ook wel over horizontale schaalbaarheid.

General purpose computing

Veel lastiger wordt het gebruik van many-core processoren op de desktop en voor general purpose toepassingen. Het parallelisme ligt daar bepaald niet voor het oprapen. Dergelijke toepassingen staan bovendien het grootste deel van hun tijd op I/O te wachten. In die tussentijd kunnen mooi instructies van andere processen wordt uitgevoerd. Dat is het werk van de scheduler van het operating system. Zo draaien tijdens het bewerken van een document op de achtergrond bijvoorbeeld de virusscanner, de mail-client en allerlei administratieve taken gewoon mee. De belangrijke vraag is dan ook wat many-core processoren – in de nabije toekomst eerst met tientallen, later met honderden kernen – ons te bieden hebben als het om general purpose computing gaat. Wie zit te wachten op de zoveelste grafische interface met allerlei onzintoeters en bellen die de productiviteit alleen maar in de weg zitten? Wat willen we

AMD Phenom Quad Core 3





AMD Phenom Quad Core 1

nog meer nu onze WYSIWYG-tekstverwerker na elke afzonderlijke toetsaanslag het hele document opnieuw opmaakt? Hoe wil je dat nog versnellen of uitbreiden om daarmee de aanschaf van een many-core processor te rechtvaardigen?

Virtualisatie

De enige toepassing die nu zonder aanpassingen direct geschikt lijkt voor de many-core architectuur, is virtualisatie. Daarmee wordt het aantal processen immers verveelvoudigd. Omdat al die virtuele machines volkomen onafhankelijk zijn, hoeft er ook niets in de code van de toepassingen gesynchroniseerd of aangepast te worden. Maar die onafhankelijkheid is tegelijk het zwakke punt van virtualisatie. Zaken die nu door alle processen via het eronder gelegen operating system worden gedeeld – softwarelibraries en de verdeling van hardwareresources – worden voor elke virtuele machine opnieuw gekopieerd. Dat maakt dat gevirtualiseerde systemen bijzonder inefficiënt met hun geheugenbandbreedte omgaan. Het

Sun UltraSparcT1



hele besturingssysteem wordt immers meerdere malen gekopieerd, en dat terwijl het geheugen al de grote bottleneck is in de hedendaagse architecturen. VMware probeert deze inefficiëntie eruit te halen met behulp van hun Transparant Page Sharing. Daarbij worden verschillende geheugenpagina's met elkaar vergeleken om te zien of ze overeenkomen. Blijkt dat inderdaad het geval te zijn, dan worden ze op elkaar gemapt. Omwille van de veiligheid wil de virtualisatie-leverancier echter geen speciale faciliteiten in het operating system inbrengen om deze taak te vergemakkelijken. De virtuele machines moeten vooralsnog volkomen transparant blijven.

Programmeurs

Met dit alles is het gebruik van de komende generaties processorarchitecturen vooral een probleem voor de softwareontwikkelaars geworden. Intel doet wel zijn best door het ontwikkelen van nieuwe programmeertalen en het uitbreiden van bestaande talen. Hetzelfde geldt bijvoorbeeld voor Microsoft, dat zijn ontwikkelomgeving uitbreidt met de Parallel Extensions. Denk bij dit soort uitbreidingen dan vooral aan betere thread-libraries en parallele versies van bestaande flow-instructies. Bovendien worden universiteiten bijvoorbeeld via fondsen van Intel gestimuleerd om hun studenten op te leiden in het programmeren van multi-threaded toepassingen en om onderzoek te doen naar het gebruik van many-core systemen. AMD startte onlangs het Multicore Threadfest, waarbij elk kwartaal kleine geldprijzen worden toegekend aan slimme multi-threaded programmaatjes. Uiteindelijk zullen het dus de programmeurs zijn die in hun algoritmen op zoek moeten gaan naar parallellisme, en die vervolgens zullen moeten vertalen naar nieuwe parallele datastructuren en functies. «