

We eisen steeds meer van onze computers. Ze draaien zware simulatieprogramma's. Ze ploeteren om foto's en video te bewerken. En dat trekt één processor niet meer. Daarom zetten we er twee of nog meer in. De prestaties nemen echter niet evenredig toe met het aantal processoren. "Je kunt niet zomaar een som maken: maal twee, maal vier."

Multicore-technologie

Interview met CTO Enno Romkema

Multicore is een buzzword, maar wat is het nu precies? En waarom hebben we het nodig?

De snelheid van pc-processoren blijft al enkele jaren hangen rond de drie GHz. Dat heeft te maken met fysische beperkingen. Naarmate de frequentie toeneemt, gaat stroom zich anders gedragen. Je krijgt steeds meer storende effecten. Maar ook warmteontwikkeling speelt een rol. De fabricagetechnieken verbeteren nog steeds dus kunnen we ze wel kleiner maken maar niet sneller. Om toch meer performance te krijgen, zet de industrie meerdere processoren bij elkaar. Dat is waarschijnlijk het bekendste voorbeeld van multicore-technologie. Maar het is veel breder dan veel mensen denken. Multicore wordt ook volop toegepast in industriële systemen.

Hoe werkt multicore-technologie?

Hoe weet elke kern wat hij moet doen? Dat is nu juist een probleem van multicore-technologie. Het vraagt flink wat communicatie en coördinatie om een multicore-systeem goed te laten werken. Vergelijk het met het schrijven van een verhaal. Stel dat je een artikel met zijn tweeën schrijft, dan moet je heel veel afstemmen. Doe jij de even vragen, dan doe ik de oneven vragen. In welke volgorde doen we het? Als je in je eentje schrijft, bepaal je zelf de vorm en de volgorde. Maar nu ga je het met zijn tweeën doen en moet het ook nog eens twee keer zo snel klaar zijn. Dat is ongeveer wat er gebeurt bij een dualcore-systeem. Twee CPU's zijn op gelijke wijze aangesloten op het hoofdgeheugen en elke CPU kan op dezelfde

manier van dat geheugen gebruik maken. Dit is een symmetrisch systeem.

Zijn er ook asymmetrische systemen?

Jazeker, en die gebruiken we al jaren. Multicore-technologie is dus niets nieuws. In een asymmetrisch systeem heeft elke core een dedicated functie. Ergens zit een moeder-CPU waar alles samenkomt en die de boel coördineert. De mobiele telefoon zit zo in elkaar, met een baseband-processor, een applicatieprocessor, enzovoorts. Elke functie draait op een aparte CPU. Maar dat is niet wat we anno 2008 doorgaans onder multicore-technologie verstaan. Want dat is: meer CPU-cycles realiseren met meerdere cores om daarmee aan het probleem van stagnerende performance te ontsnappen.

Loopt de performance gelijk op met het aantal cores? Is een quadcore inderdaad vier keer sneller?

Nee, zo werkt het helaas niet. Zowel de hardware als de software moet geoptimaliseerd worden voor multicore. Qua hardware is de bandbreedte van het geheugen de beperkende factor. Dat is de hoeveelheid data die per tijdseenheid in en uit het geheugen kan. Je boekt alleen winst als de CPU's echt hun eigen gang kunnen gaan. Als je van één naar twee CPU's gaat, heb je een performance-toename van minder dan twee. En als je van twee naar vier gaat is er bijna geen winst meer. Het geheugen vormt de bottleneck.

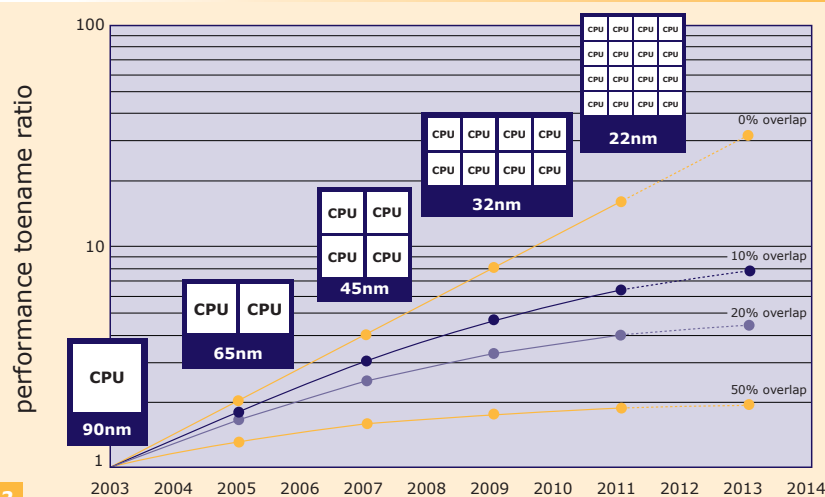
Is dat op te lossen met extra geheugen?

Ja, dat gebeurt ook. Elke core krijgt een

Je boekt alleen winst als de CPU's echt hun eigen gang kunnen gaan.



1 2



1. Multicore betekent niet altijd tijdswinst. 2. De wet van Amdahl geeft goed weer dat tijdswinst afhankelijk is van opeenvolgende software activiteiten.

privégeheugen of cache. Maar die caches introduceren overhead. Er kunnen verschillen ontstaan tussen het centrale werkgeheugen en de caches van de afzonderlijke kernen. Omdat deze verschillen moeten worden bijgewerkt neemt de voorspelbaarheid van de snelheid van de software af. Dat geeft vooral problemen bij realtime processen. Naarmate er meer CPU's komen, nemen die problemen alleen maar toe. Alleen maar meer geheugen erbij zal dan ook geen oplossing zijn.

Wat moet er in de software gebeuren om het maximale uit een multicore-systeem te halen?

Je moet de software opknippen, elke CPU zo lang mogelijk zijn eigen ding laten doen en het eindresultaat weer samenvoegen. Dat heet partitioneren. Software voor de pc is generiek geschreven en daar wordt geen rekening gehouden met de architectuur van de CPU. De winst van een multicore in een pc hangt daarom af van de applicatie. Een industrieel apparaat is gemaakt voor een strak omlijnde toepassing. Daar kun je specifieke taken aan een core toewijzen. Industriële besturingssystemen kunnen dat ondersteunen, in tegenstelling tot Windows. Zo kun je een optimalisatie bereiken die je op een pc nooit haalt. Dat maakt multicore-technologie zo interessant voor technische toepassingen.

Heb je een nieuwe taal of speciale tools nodig om software voor multicore-technologie te ontwikkelen?

Nee. We kunnen nog steeds de bestaande gereedschappen gebruiken om goede algoritmen te ontwerpen en taken over meerdere CPU's te verdelen. Maar je moet je verstand gebruiken. Er is geen universele tool voor alle problemen.

Voor welke toepassingen is multicore-technologie vooral geschikt?

Multicore-technologie heeft het meest succes in

toepassingen met veel data die vooraf al goed kan worden gepartitioneerd. Beeldbewerking is een goed voorbeeld. Vaak kunnen 2D-beelden in stukken worden verdeeld waarbij ieder stuk door een eigen CPU kan worden verwerkt. De uitkomsten kunnen dan later weer bij elkaar gevoegd worden. Het wordt helemaal een uitdaging als met multicore meerdere 2D-beelden tot een 3D-beeld moet worden geconstrueerd. Daar is heel veel rekenkracht voor nodig. Een pixel wordt dan een "voxel": een volume-cel, een driedimensionale pixel. En elke voxel moet wel worden berekend.

Wat vindt u van multicore?

Het is leuk en nuttig, maar vergeet niet wat nou eigenlijk je probleem is. Wat gaat er niet snel genoeg? Is het probleem wel te partitioneren? Kunnen we dat niet op een andere manier oplossen? Asymmetrisch bijvoorbeeld. Met een FPGA. Als er beeld bewerkt moet worden, waarom doen we dat dan niet in hardware in plaats van software? Enzovoorts. Een system engineering benadering om het probleem op te lossen en een beetje nuchterheid zijn op zijn plaats. Multicore-systemen lossen een klein deel van het probleemveld op. Voor allerlei andere problemen moet je kijken wat de beste oplossing is. En dat moet je blijven doen. Multicore-technologie is geen doel op zich. Bovendien geeft het meer complexiteit en de winst in performance valt vaak tegen. Je kunt niet zomaar een sommetje maken: maal twee, maal vier. Zo werkt het niet. Vergelijk het met het verven van de muren van een kamer en je hebt maar één bak waarin je de roller kunt dopen. Dat kun je goed met z'n tweeën doen. Dan loop je elkaar niet snel in de weg. Met zijn drieën is die kans al groter dat je op elkaar moet wachten en met vier zul je zeker op elkaar moeten wachten. Er van uitgaande dat de kamer überhaupt groot genoeg is voor vier personen.

Een system engineering benadering om het probleem op te lossen en een beetje nuchterheid zijn op zijn plaats.