Supercomputere – mange bække små...

Af **Brian Vinter,** DIKU

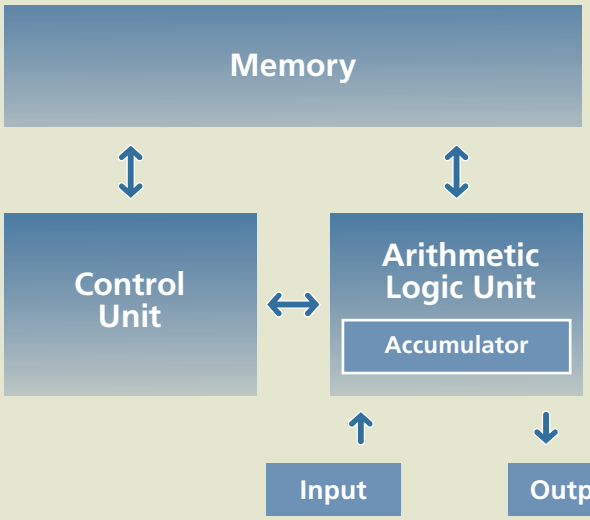
Hastigheden på en computers centrale regneenhed, CPU'en, har altid været en vigtig konkurrenceparameter for leverandørerne. Med hurtigere computere får vi bedre vejrudsigter, og vi kan simulere processer i fysik og kemi, som vi ellers måtte bygge kostbare eksperimenter for at gennemskue. Når man søger på Google, leder deres computere i 100 PB (100.000.000.000.000.000 bytes) af information, som de har kopieret fra internettet. For at det skal kunne ske så hurtigt, som det gør, må de mange data være gennemregnet på forhånd – så samtidig med at informationen på internettet stiger, stiger behovet for hurtigere computere tilsvarende. Tidligere beroede computerens hastighed på taktfrekvens, og vi så hastigheden på en pc gå fra 5 MHz til 4 GHz fra 1979 til 2004, dvs. 800 gange hurtigere taktfrekvens på blot 25 år. I dag markedsføres CPU'er med flere og flere kerner, hvilket løst kan oversættes til "Køb 1 CPU, og få 2, 4, 8 eller endnu flere". Men begejstringen over disse mange-kerne-CPU'er er til at overskue, hvorfor?

Computerens taktfrekvens

Computerens hastighed angives ofte som dens taktfrekvens. CPU'en fungerer på den måde, at den har et lille ur, der slår takten for dens fremgang; hver gang uret slår, kan CPU'en begynde en ny instruktion. Når en computer har en 3 GHz CPU, betyder det altså, at CPU'ens ur slår 3 milliarder gange i sekundet.

Den grundlæggende computermodel

Helt basalt fungerer en computer ved, at den centrale regneenhed (Central Processing Unit eller CPU'en) udfører instruktionerne i et program en ad gangen, se figur 1. Instruktionerne til programmet ligger i computerens hukommelse, og det samme gør de data, som programmet regner på. Den mest udbredte computermodel hedder von Neumann-modellen, og her ligger både instruktionerne til programmet og de data, computeren regner på, i den samme hukommelse. I von Neumann-modellen består det, vi i dag tænker på som en CPU, af to dele, dels en kontroldel, der styrer udførelsen af et program, dels en regnedel. Der findes også en model, hvor instruktioner og data ligger i hver sin hukommelse, kaldet Harvard-modellen, men den ses sjældent anvendt i generelle computere.



Figur 1: Den oprindelige von Neumann-computermodel, hvor CPU'en er opdelt i hhv. en kontroldel (Control Unit) og en regnedel (Arithmetic Logic Unit). CPU'en læser instruktioner fra hukommelsen (Memory). Data læses og skrives fra samme hukommelse.

I dag har en almindelig computer ofte to eller flere CPU-kerner, hvilket betyder, at den har flere CPU-lignende enheder, som deler en fælles hukommelse og derfor skal samarbejde for at løse en beregningsopgave.

Moderne forskning kræver mange FLOPS

Måleenheden for en computers regnekraft hedder FLOPS, floating point operations per second, eller på dansk: antal beregninger på kommatal pr. sekund. Vi har brug for meget stor regnekraft for at lave moderne videnskab, hvor computere bruges til at beregne kemiske bindinger i medicin, styrken af og elektriske egenskaber ved nanopartikler, effekten af en ny vindmølle eller jordens fremtid i form af klimamodeller – bare for at nævne nogle få anvendelser, som kræver meget store supercomputere.

Måleenheden for en computers regnekraft hedder FLOPS

Verdens hurtigste kommatal-computer i år 2010 har en teoretisk hastighed på 2,3 billiarder FLOPS, også skrevet som 2,3 PFLOPS, og en målt hastighedsrekord på 1,76 PFLOPS. Denne computer kan altså lave 1,76 billiarder beregninger pr. sekund.

Men det er ikke kun verdens supercomputere, der bliver stadig hurtigere. Det samme gælder almindelige pc'er, som de fleste mennesker i dag har mindst én af. I dag er fire eller otte CPU-kerner og mere end 30 GFLOPS ganske almindeligt. Faktisk har en pc, der er beregnet til spil, en speciel Graphics Processing Unit (GPU) til at beregne grafik. Denne GPU har ofte 240 kerner og regner med så meget som 1,4 TFLOPS.

FLOPS

FLoating point **O**perations **P**er **S**econd eller på dansk: Antal beregninger på kommatal pr. sekund. En beregning kan fx være en addition, en subtraktion, en multiplikation eller en division af to kommatal.

1 FLOPS = 1 beregning pr. sekund
1 MFLOPS = 1 million, 10^6 beregninger pr. sekund
1 GFLOPS = 1 milliard, 10^9 beregninger pr. sekund
1 TFLOPS = 1 billion, 10^{12} beregninger pr. sekund
1 PFLOPS = 1 billiard, 10^{15} beregninger pr. sekund

Definitionen af FLOPS forudsætter, at de kommatal, der benyttes, kun har et begrænset antal cifre, dvs. at det samlede antal cifre før og efter kommaet er begrænset. Der findes to typer af FLOPS, single precision (SP) og double precision (DP). Et DP-kommatal har dobbelt så mange mulige cifre som et SP-kommatal. En DP er dobbelt så stor som en SP, og når FLOPS bruges som måleenhed, er det altid DP, man bruger, medmindre man skriver, at det er SP. Oftest er en CPU's SP-ydelse dobbelt så stor som dens DP-ydelse.

Verdens hurtigste supercomputer

Den til enhver tid hurtigste computer i verden kan findes på www.top500.org.
Lige nu (år 2010) er det den amerikanske computer "Jaguar", der topper listen. Denne computer har 224.162 kerner, der tilsammen kan levere en faktisk regnekraft på 1,76 PFLOPS, mens den teoretiske hastighed er 2,3 PFLOPS, altså lige over 10 GFLOPS pr. kerne.

> ***Den til enhver tid hurtigste computer i verden kan findes på www.top500.org***

Moores lov forudsiger udviklingstakten i CPU'ers hastighed

At vi kan bygge stadig hurtigere computere, skyldes en teknologisk udvikling, der er formuleret i Moores lov, som ikke er en lov i naturvidenskabelig forstand, men snarere en slags tommelfinger-regel, som siger, at størrelsen på en transistor på et integreret kredsløb som fx en CPU halveres hver 18. måned.

Tidligere forudsagde Moores lov også, at CPU'ernes taktfrekvens blev fordoblet hver 18. måned; det skyldtes, at transistorerne jo var blevet halvt så store, og derfor kunne computeren skifte tilstand på den halve tid. Desværre medførte dette, at CPU'erne også brugte stadig mere strøm, hvilket igen betød, at de blev hurtigere ophedet.

Moores lov gælder stadig

Da Intel i 2004 annoncerede, at de ikke ville bygge en 4 GHz Pentium 4 CPU, bekendtgjorde de samtidig, at hvis de skulle fortsætte som hidtil, ville en CPU i 2007 have 16 GHz taktfrekvens og blive 6.000 grader varm, mens den kørte. Derfor har vi ikke siden 2004 set nogen nævneværdig stigning i taktfrekvensen på en CPU, og computeren bliver derfor ikke længere hurtigere til at køre programmer på en enkelt CPU-kerne.

Men Moores lov gælder stadig. Det betyder, at en CPU, der i dag har 500 millioner transistorer, om 1½ år vil have 1 milliard transistorer og om 3 år 2 milliarder transistorer. For at lave en kommatal-beregning, fx addition af to tal, skal vi bruge ca. 1.000 transistorer, så en CPU med 1 milliard transistorer kunne, hvis den udelukkende skulle addere, foretage 1 million beregninger pr. taktslag. En hurtig CPU har i dag næsten 4 milliarder taktslag i sekundet, så den ville altså kunne lave næsten 4 millioner milliarder eller 4 billiarder beregninger pr. sekund (4 PFLOPS). Nu skal en CPU jo ikke kun addere men også kunne lave mange andre ting, som der skal bruges transistorer til. Et rimeligt gæt ville være, at en milliard transistorer skulle regne med 1 TFLOPS. Desværre er virkeligheden, at vi bruger ganske få af transistorerne på en CPU til at regne med. En almindelig CPU med en milliard transistorer leverer snarere 100 GFLOPS, mens en GPU kommer tæt på 1 TFLOPS med 1 milliard transistorer.

Hvorfor kan man ikke bare skalere op?

Moores lov forudsiger, at en computer får stadig mere regnekraft, for mens taktfrekvensen ikke stiger, så får hver computer stadig flere kerner. Hvis en kerne kan regne med 10 GFLOPS, så burde 1.000 kerner vel kunne regne med 10 TFLOPS? Desværre er det ikke så let, idet man for at kunne udnytte 1.000 kerner må omskrive programmet til at kunne deles op i mindre dele, som så kan køre samtidig – man taler om at parallelisere arbejdet.

Det lugter af et simpelt regnestykke fra folkeskolen: Hvis én mand kan grave en grøft på en time, hvor hurtigt kan to mænd så gøre det? En halv time ville vi nok hurtigt gætte på, men hvad med 1.000 mænd? Kan de gøre det på 3,6 sekunder? Det er klart for enhver, at man ikke kan grave en grøft på 3,6 sekunder – de mange mennesker ville simpelthen stå i vejen for hinanden. Sådan er det også, når vi skal parallelisere computerprogrammer; det kan være meget vanskeligt, selv for de letteste problemer.

Lad os prøve et meget simpelt forsøg. Vi vil gerne finde summen af en masse tal, helt præcist 10.240 tal. Lad os sige, at vi har en computer med 1.024 kerner, dvs. at hver kerne skal finde sum-

men af ti tal. Dette kan gøres på ni taktslag ved at lægge de to første tal sammen, lægge summen til det tredje tal til osv. Derefter har vi 1.024 delsummer, og dem må vi så lægge sammen. Hvis en kerne gør det, så vil det tage 1.023 taktslag, og det er jo meget mere, end vi brugte på at få opgaven ned fra 10.240 til de 1.024 tal, der skal summeres. De første 90% af opgaven vil altså gå meget hurtigere end de sidste 10%, så det må vi kunne gøre bedre!

Løsningen er selvfølgelig at dele opgaven op, således at 512 kerner lægger tallene sammen to og to, derefter 256, 128, 64, 32, 16, 8, 4, 2 og til sidst 1 kerne for den sidste sum. Det betyder, at vi må bruge yderligere ti taktslag til at løse opgaven. Det betyder, at vi bruger dobbelt så lang tid på at løse opgaven, som man skulle tro, når man dividerer opgavens størrelse med antallet af kerner i vores computer. Dette er naturligvis bare overfladen af problemet. At lægge tal sammen er den letteste opgave, vi kan forestille os, og den lader sig jo forholdsvis let parallelisere. Der er mange andre og mere værdifulde opgaver, der er langt vanskeligere at parallelisere.

Hukommelsen er også for langsom

Desværre er problemet med de mange kerner ikke den eneste udfordring, som dataloger står over for, når computere skal levere mere regnekraft. Det helt store problem ligger i selve von Neumann-modellen, nemlig i, at CPU'en hele tiden skal læse fra hukommelsen – og desværre er hukommelsen meget langsommere end CPU'en. Hukommelsen i en computer er opbygget i en langsommere teknologi for at kunne presse så meget hukommelse ind som muligt. En almindelig pc har måske 4 GB af typen dynamisk hukommelse. Hvis man skulle bruge en hurtigere type, kaldet statisk hukommelse, kunne den samme computer kun have plads til ca. 256 MB hukommelse eller 16 gange mindre hukommelse. En hurtig hukommelse kan CPU'en læse fra på 15 nanosekunder, altså 15 milliardtedele af et sekund. Men en hurtig CPU-kerne kører ved 3 GHz, dvs. at den har en taktfrekvens på 3 milliarder instruktioner pr. sekund. Det betyder så, at en kerne har tre taktslag hvert nanosekund, men hukommelsen kan altså kun tilgås hvert 15. nanosekund. Med andre ord er en CPU-kerne 45 gange hurtigere end den hukommelse, den skal læse fra. Faktisk er CPU-kernen opbygget sådan, at den kan køre fire instruktioner for hvert taktslag, så CPU-kernen er faktisk 180 gange hurtigere end hukommelsen. Hvis vi har 12 kerner på en CPU, er den altså 12 x 180 = 2.160 gange hurtigere end hukommelsen. Hukommelsen er 2.160 gange langsommere, og en CPU skal både læse instruktioner og data fra hukommelsen, så vi har altså et enormt problem med den langsomme hukommelse.

Den klassiske løsning på problemet med den langsomme hukommelse hedder cachehukommelse. Idéen med cachehukommelse er at have en hurtig, men relativt lille hukommelse, der ligger meget tæt på CPU'en, og så have en kopi af en del af den langsomme hukommelse liggende i den hurtige hukommelse. Problemet med cachehukommelsen er, at den kun virker, når computeren kan gætte, hvilke data CPU'en skal bruge, eller hvis programmøren har skrevet programmet på en måde, så man ved, at CPU'en kan gætte, hvilken del af hukommelsen der skal bruges. Moderne CPU'er er udstyret med ganske meget cachehukommelse i et forsøg på at forhindre CPU-kernerne i at vente hele tiden, og en stadig større del af de transistorer, vi kan bruge i en CPU, bliver brugt til cachehukommelse. Det kan synes som lidt af et ressourcespild at bruge gode transistorer på noget så "kedeligt" som at kopiere hukommelsen. Derfor er der også en række nye tendenser i CPU-design, som vi vil se på sidst i artiklen.

Hvor meget hurtigere kan computeren blive?

Når vi skal bygge meget store supercomputere, må vi bruge mange CPU-kerner. De største computere har mere end 200.000 CPU-kerner, og selvom elektronikdelene er ganske små, fylder 200.000 CPU'er stadig temmelig meget, ofte så meget som en hel sportshal. CPU'erne må kommunikere over ganske lange afstande, og der har vi det næste problem: lysets hastighed. CPU'erne kommunikerer med strøm via et netværk, og strøm bevæger sig noget langsommere end lys (afhængigt af det materiale, netværket er lavet af).

For at gøre det let vil vi i dette afsnit antage, at computere kan snakke sammen med lysets hastighed. Det lyder temmelig fjollet at sige, at lyset bevæger sig langsomt – vi siger jo netop "med lysets fart", når noget går stærkt. Men i denne sammenhæng er lyset faktisk ret langsomt. Lysets hastighed er ca. 300.000 km pr. sekund, det samme som 1 fod eller 30 cm pr. nanosekund. Vi har tidligere regnet ud, at en CPU-kerne ved 3 GHz og 4 instruktioner pr. taktslag kan udføre 3 * 4 = 12 instruktioner pr. nanosekund. Dvs. at mens lyset bevæger sig 1 m, kan en sådan CPU udføre 40 beregninger på den samme tid. Hvis vi går ud fra, at en computer er 50 m lang, kan en CPU altså udføre 4.000 instruktioner på den tid, det tager at sende en besked fra den ene ende af computeren til den anden og så få et svar tilbage. Den faktiske tid er pga. elektronikken omkring 1.000 gange langsommere. Hvis en CPU skal bede en anden CPU gøre noget af arbejdet for sig, skal det arbejde altså være på mindst 4 millioner instruktioner – ellers bruger den mere tid på at kommunikere end den tid, som CPU'en sparer ved ikke at gøre arbejdet selv. Disse tal er alle for én computer i ét lokale; hvis man i stedet vil bruge computere, der er forbundet via internettet, til at samarbejde, så er det hele igen en million gange langsommere. Kun rigtig store opgaver kan sendes til en anden computer, hvis man skal spare tid på det.

> *I denne sammenhæng er lyset faktisk ret langsomt*

Forhold mellem afstand og antal CPU-instruktioner

I tabellen herunder antages det, at vi kan kommunikere med lysets hastighed, og at vi har en 3 GHz CPU, der udfører 4 instruktioner pr. taktslag.

Afstand	Tid for lyset	CPU-instruktioner
1 fod	1 nanosekund	12
1 m	3,3 nanosekunder	40
1 km	0,0000033 sekunder	40.000
Jorden rundt ved ækvator (40.066 km)	0,13 sekunder	1.602.640.000

Tre tendenser i CPU-design

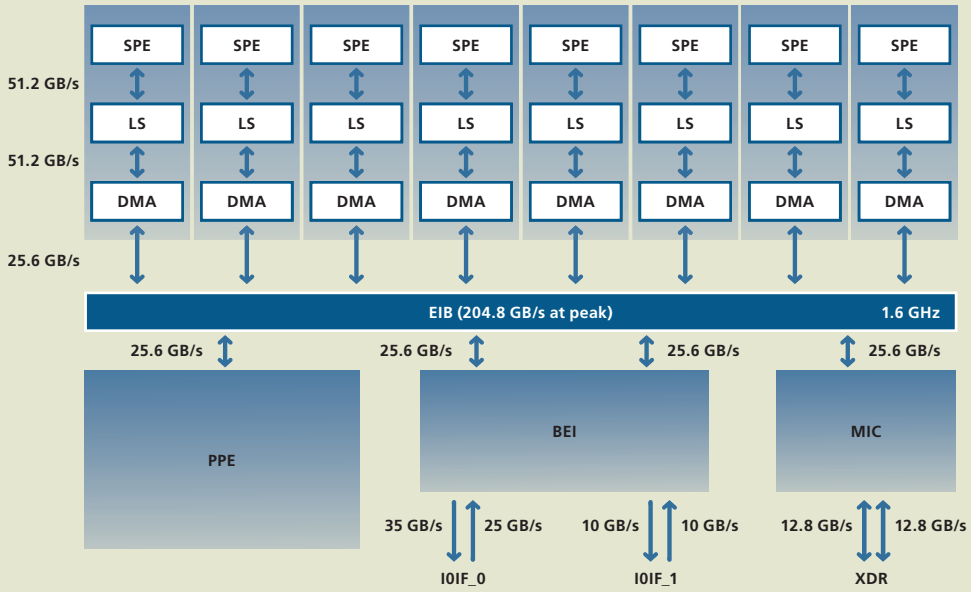
Dataloger og ingeniører står over for en stor udfordring. Moores lov forudsiger, at vi får stadig flere transistorer på en CPU; det giver flere kerner, der kan lave beregninger samtidig. Derudover er hukommelsen langsom, og når CPU'er skal tale sammen, er selv lysets hastighed langsom. Problemet med den langsomme hukommelse er det mest umiddelbare og derfor det problem, der arbejdes hårdest på at løse. Der synes at være tre generelle tilgange til problemet:

1. Gøre hukommelsen hurtigere
2. Gøre hukommelsen hurtigere i gennemsnit
3. Gøre CPU'en relativt langsommere.

IBM har med Cell-BE-CPU'en, som man bl.a. finder i PlayStation 3-konsollen, arbejdet med at gøre hukommelsen hurtigere. I stedet for at bruge den klassiske cachehukommelse, hvor CPU'en forsøger at gætte, hvilke data programmet skal bruge, så har CELL-BE+ en lille, men hurtig hukommelse til hver CPU-kerne. Denne hukommelse er så den eneste, som kernen kan bruge, men kernen har derudover en lille hjælpekomponent, der kan overføre blokke af hukommelse mellem den lille, hurtige og den store, langsomme hukommelse.

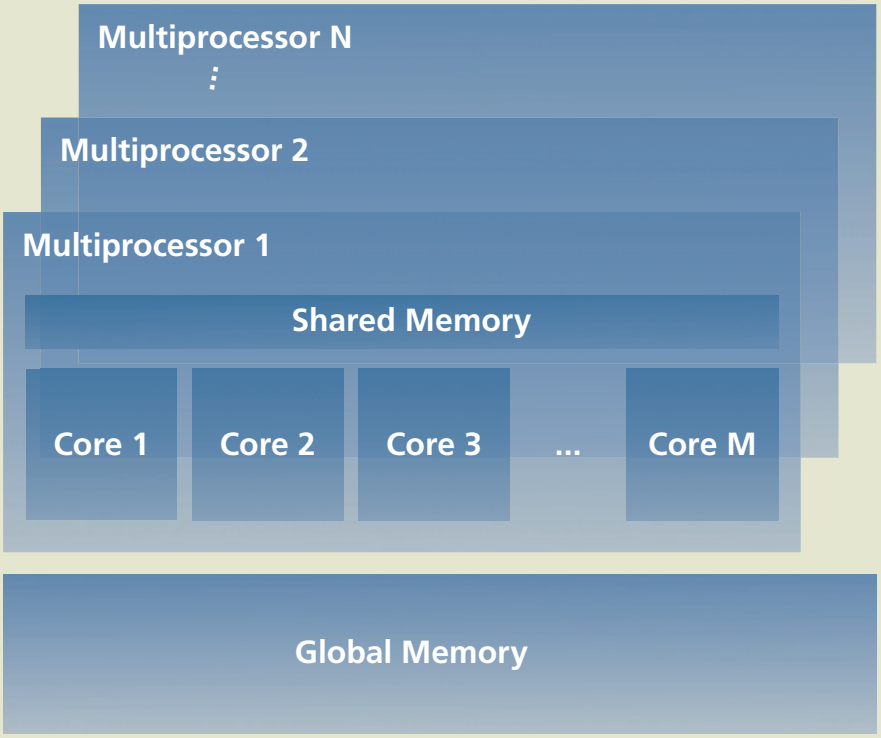
IBM CELL-BE+

CELL-BE+ CPU'en har ni kerner: én klassisk CPU-kerne og otte specielle CPU-kerner med hver deres hukommelse. De ni CPU-kerner er forbundet af et internt, meget hurtigt netværk. Hver enkelt CPU kan regne med mere end 100 GFLOPS. Spillekonsollen PlayStation 3 bruger en ældre og mindre version af den samme CPU, CELL-BE.



NVidia Fermi

Tesla, som er den største processor fra NVidia lige nu (2010), fås med 448 kerner, der kan levere 515 GFLOPS regnekraft. NVidia har tidligere fokuseret på grafikprocessorer, og Tesla er den første CPU, de udvikler, der er tiltænkt videnskabelige beregninger.

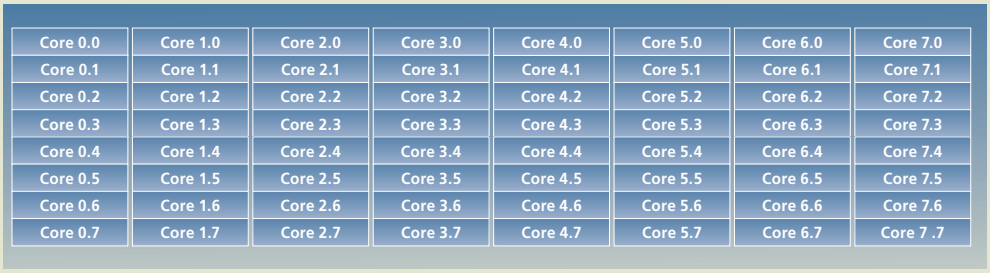


Firmaet NVidia, som er bedst kendt for at lave grafikprocessorer, GPU'er, er også gået ind på markedet for videnskabelige beregninger. Deres løsning er at gøre hukommelsen hurtigere i gennemsnit. Idéen er lidt den samme som at bruge en lastbil til at flytte med frem for en personbil. Lastbilen kan ikke køre hurtigere end personbilen – ofte omvendt – men til gengæld kan den have meget mere med. I en computer svarer dette til, at man frem for at nøjes med 8 bytes, når man læser fra hukommelsen, i stedet henter 448 bytes eller endnu mere. Dette fungerer selvfølgelig kun, når man skal bruge alle de bytes, der hentes. Det problem løser NVidia ved at dele de mange bytes, der hentes, op imellem mange, op til 448, CPU-kerner. Dermed bliver det op til programmøren at skrive et program, der kan køre parallelt på så mange kerner, at man udnytter størrelsen på den hukommelse, der hentes.

Den sidste model, at gøre CPU'en relativt langsommere, virker måske fjollet, men når CPU'en er for hurtig til hukommelsen, er der kun to muligheder tilbage: at gøre hukommelsen hurtigere eller at gøre CPU'en langsommere. Sun bruger den sidste model, og idéen er meget simpel. Hvis programmøren alligevel skal skrive et parallelt program, så kan programmet jo skrives til mange flere kerner, end der faktisk er på CPU'en. I Suns Niagara-processorlinje deler man en CPU-kerne op i 8 eller 16 virtuelle kerner over tid. Det fungerer på den måde, at hvis man bruger to virtuelle kerner, så vil hver af disse få hver andet taktslag, dvs. at CPU'en har fået dobbelt så mange kerner, der hver kører halvt så hurtigt. Det giver den samme totale beregningskraft, men til gengæld er hastighedsforskellen til hukommelsen for hver kerne blevet halvt så stor. Når man så gør det med 16 virtuelle kerner, bliver forskellen reduceret til 1/16; til gengæld skal en CPU med 8 fysiske kerner så programmeres, som om den havde 128 kerner.

Sun Niagara

Niagara-CPU'erne fra Sun, T1, T2 og T3, bruger virtuelle CPU-kerner til at få programmerne til at køre langsommere. Selvom det lyder ulogisk, fungerer det fint, fordi den samlede regnekraft i CPU'en ikke bliver mindre. Til gengæld virker hukommelsen ikke så langsom for de enkelte kerner.



Nogle friske forskningsresultater

For programmører er de flere forskellige tilgange til at løse hastighedsproblemet faktisk en dårlig nyhed. Det er ikke til at forudsige, hvilken teknologi der viser sig at være den bedste, og de forskellige løsninger kræver alle forskellige måder at programmere på. Det betyder, at hvis man som programmør beslutter sig for at skrive til modellen med de virtuelle kerner, og den teknologi så viser sig ikke at overleve, så kan programmerne ikke umiddelbart bruges til de næste generationer af CPU'er. Et af forskningsprojekterne på DIKU går ud på at løse dette problem og forsøger at udvikle en programmeringsmodel, der passer til alle tre løsninger.

Idéen er at tage udgangspunkt i et eksisterende programmeringssprog – i dette tilfælde Python – og så sprede opgaverne, som et program specificerer, ud over de CPU'er, der er til rådighed.

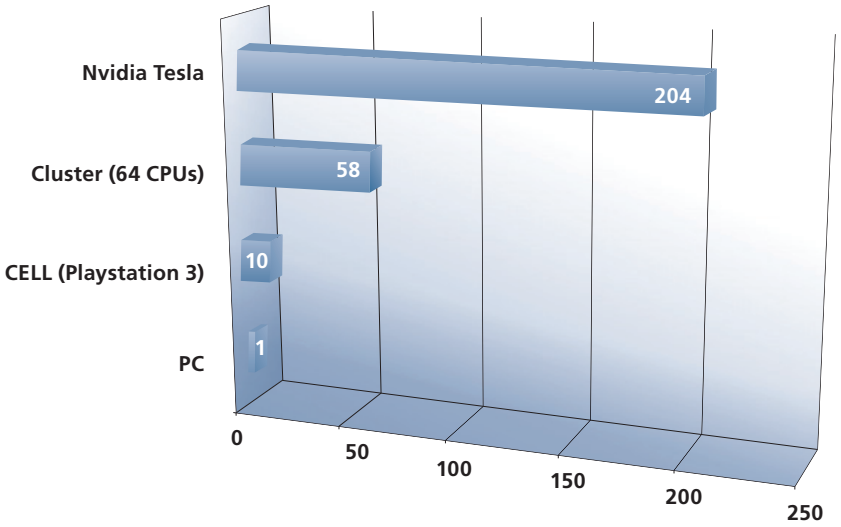
Projektet er i skrivende stund i sin opstart, men der er allerede kommet nogle resultater. Testprogrammet i figur 2 er et meget simpelt program, der forsøger at beregne værdien af π med en metode, der kaldes Monte Carlo-simulering, og programmet er meget kort:

```
def McPi(s=samples):
    x,y=(random(s),random(s))
    (x,y)=(x*x,y*y)
    x+=y
    return sum(less_equal(x, 1.0))*4.0/s
```

Figur 2: Monte Carlo-simulering.

Denne type opgave er meget let at parallelisere, og man kan da også se på resultaterne i figur 3, at de er rigtig lovende. Tallene kaldes speed-up. Vi har altså en almindelig kraftig pc, der kører almindelig Python, som får basisnummeret 1. En PlayStation 3 er 10 gange hurtigere, en klyngecomputer med 64 kerner er 58 gange hurtigere, og et Tesla-kort fra NVidia er 204 gange hurtigere end den almindelige pc. Alle fire maskiner kører præcis det samme program, så programmøren skal ikke ændre noget for at skifte teknologi.

Men det betyder også, at programmøren kan blive fri for at bekymre sig om, hvilken teknologi der i sidste ende viser sig at være den bedste, fordi programmer skrevet i DIKUs system kører effektivt på alle tre typer uden nogen form for ændringer.



Figur 3: Testresultater foretaget med Monte Carlo-metoden. Grafen viser den relative merydelse i forhold til en almindelig pc, som de parallelle maskiner kan præstere.

Supercomputeren på Københavns Universitet

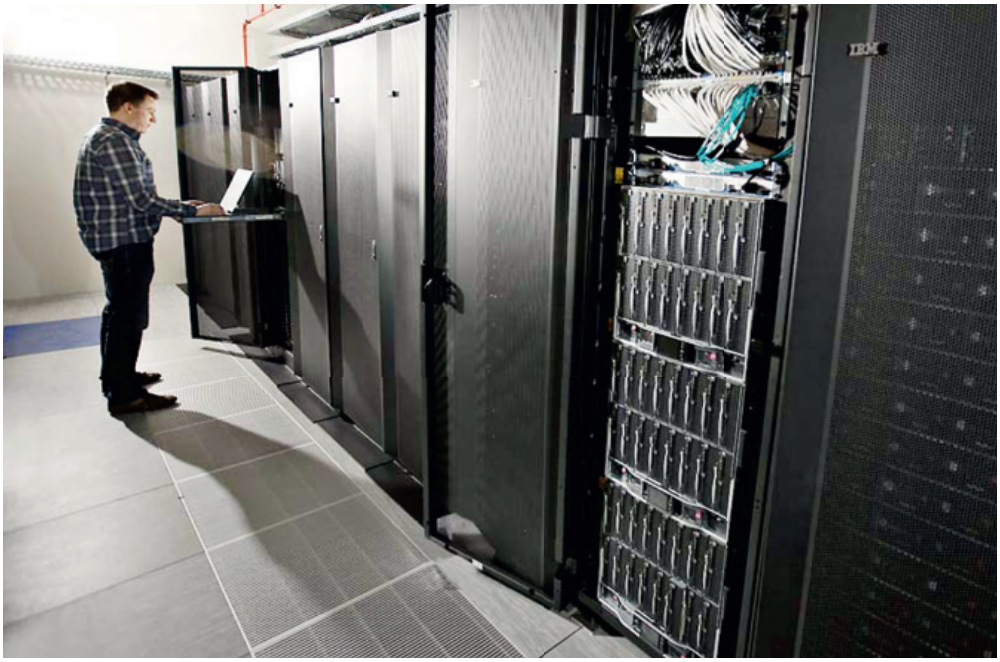
Københavns Universitet råder over en supercomputer, der har 4.560 CPU-kerner og kan regne med en hastighed på 45 TFLOPS. Computeren anvendes af såvel eScience-centeret (et tværfagligt forskningscenter på KU, der tillige uddanner kandidater og ph.d.er i eScience) som flere af de individuelle institutter under Det Naturvidenskabelige Fakultet. Computeren anvendes til en række meget forskellige forskningsprojekter, der har det tilfælles, at de kræver ekstrem regnekapacitet.

Eksempelvis bruges supercomputeren af Niels Bohr Institutets forskere til at regne på de meget store mængder data, der kommer fra Large Hadron Collider-eksperimentet på CERN (i pressen bedre kendt som big bang-eksperimentet). Et andet anvendelsesområde er simulering af galaksers udvikling eller atomers bevægelser i helt små molekyler.

Derudover anvendes supercomputeren i stor stil inden for medicinsk forskning til at finde nye og bedre diagnosticeringsmetoder af en lang række sygdomme. Bl.a. kan man få computeren til at hjælpe med at stille tidlige diagnoser af sygdomme som slidgigt, KOL og brystkræft ved at analysere CT- og MR-billeder. Om anvendelse af supercomputeren i medicinsk forskning henvises desuden til Mads Nielsens artikel om machine learning. ❖

Læs mere

http://en.wikipedia.org/wiki/UltraSPARC_T3
http://en.wikipedia.org/wiki/Cell_(microprocessor)
http://en.wikipedia.org/wiki/Nvidia_Fermi



Figur 4: Supercomputeren på Københavns Universitet – der driver en række forskningsprojekter på Det Naturvidenskabelige Fakultet.

BRIAN VINTER

Brian Vinter er professor på eScience-centeret, som han har været tilknyttet siden 2007. Han er civilingeniør fra Aalborg Universitet og ph.d. fra Tromsø Universitet fra 1999. Brian har en lang akademisk karriere, bl.a. som gæsteforsker på Princeton University, CERN i Schweiz og University of Kent i England.

Han har været aktiv inden for klyngecomputer-forskning siden 1994 og har ydet betydelige bidrag til forskningen i feltet, specielt inden for distribueret delt hukommelse. Han er på DIKU også kendt som Mr. Supercomputer. Brian Vinter underviser på DIKU i fagene ekstrem multi-programmering og cluster computing. Ud af en lang liste af publikationer kan nævnes: "Next Generation Processors" (2009) og "Cycle-Scavenging in Grid Computing" (2009).

Den digitale revolution – fortællinger fra datalogiens verden
Bogen er udgivet af Datalogisk Institut, Københavns Universitet (DIKU) i anledning af instituttets 40 års jubilæum med bidrag fra forskere tilknyttet instituttet.

Redaktion:
Tariq Andersen, phd-studerende, Jørgen Bansler, professor, Hasse Clausen, lektor,
Inge Hviid Jensen, kommunikationsmedarbejder og Martin Zachariasen, institutleder.
Forsidemotiv: Foto af skulptur af Alan Turing, © basegreen lokaliseret på flickr.om/photos/basegreen
Oplag: 1000 eks.
Grafisk design og produktion: Westring + Welling A/S
ISBN: 978-87-981270-5-5