Foto: Lars Kruse

Gå ikke over åen efter data

Ved at tilrettelægge regnearbejdet rigtigt, kan man få en computer til at arbejde flere tusinde gange hurtigere. Et nyt datalogisk grundforskningscenter i Århus skal finde de allersmarteste algoritmer at gøre det med.

Af Peter F. Gammelby

Verden oversvømmes af så massive datamængder, at selv de nyeste og kraftigste computere har svært ved at følge med. Alene i løbet af 2006 blev der i verden genereret og kopieret 161 milliarder gigabytes digital information, og det tal vil være seks gange så højt i 2010, vurderer analysefirmaet IDC.

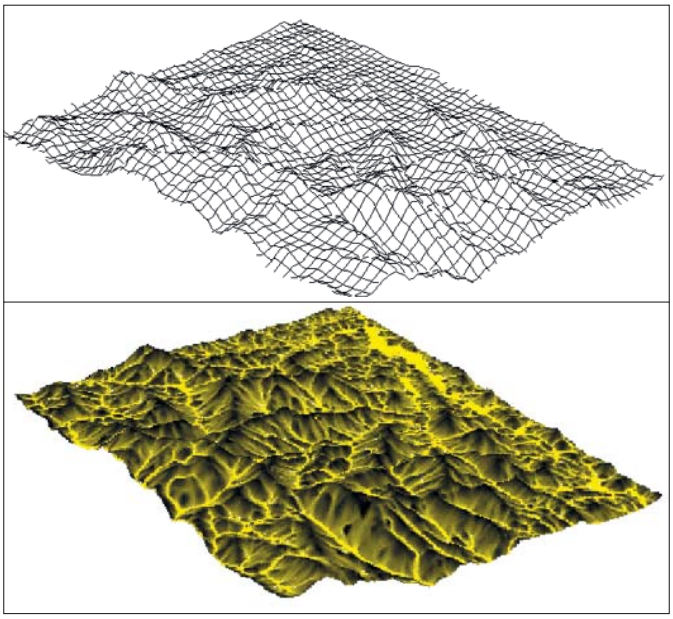
Ikke nok med, at vi alle sammen gemmer data på vore computere og uploader mange af dem til world wide web; sensorer, kameraer, telefoner og alskens elektronik omkring os medvirker til at skabe denne flodbølge af information. Flodbølgen rummer imidlertid også et hav af guldminer, både for forskere og for forretningsfolk. Det gælder blot om at finde de værktøjer, som gør det muligt at lave minedrift i så massive datamængder.

»Data-mining er ekvivalent til nano-teknologien, sådan at forstå, at vi endnu ikke kan overskue de enorme muligheder, der ligger i denne teknologi. Der er så massive datamængder og så meget at bruge dem til,« siger datalogi-professor Lars Arge fra Aarhus Universitet.

Jagt på nye algoritmer

Lars Arge er leder af det nye grundforskningscenter Madalgo (Massive Data Algorithmics) i Århus, finansieret af Dan-

← Algoritmer til at håndtere store datamængder på en computer er at sammenligne med bageopskrifter, mener professor i datalogi Lars Arge. Mange af dagens algoritmer svarer til at flyve til Australien efter æg, til USA efter mel og til Kina efter sukker, når man bager en kage.



Før tog det en computer to uger at simulere vand-flowet (nederst) på den elektroniske terrænmodel (øverst), der var skabt ved hjælp af et en laserscanner monteret på maven af et fly. Med en effektiv algoritme kunne arbejdstiden skæres ned til tre timer – på den samme computer!

marks Grundforskningsfond, hvor danske forskere sammen med kolleger fra Massachusetts Institute of Technology (MIT) i USA og Max Planck Institut für Informatik (MPII) i Tyskland skal prøve at udvikle de mest effektive algoritmer til store datamængder.

Det er nemlig ikke nok bare at øge regnekraften på de computere, som skal bruges til at analysere dataene.

Den berømte Moores Lov, som siger, at antallet af transistorer på en chip fordobles hver 18. måned, har holdt stik, siden den blev formuleret for 42 år siden. Men den eksponentielle vækst, som computernes regnekraft har gennemgået de sidste 40 år, kan ikke fortsætte med den nuværende teknologi. Der er grænser for, hvor små transistorerne kan blive, og varmeudviklingen fra processorerne er ved at være et problem for hastigheden.

Og mængden af data vokser endnu hurtigere end regnekraften gør. Desuden er massive datamængder et relativt begreb; for en lille computer kan selv små datamængder være store.

Med effektive algoritmer kan man gøre selv meget store datamængder overkommelige for almindelige computere.

Bøvlet bagning

En algoritme er en nøje og utvetydig opskrift på, hvordan en opgave skal løses, skridt for skridt. Lars Arge plejer at sammenligne det med en bageopskrift. Og når man alligevel er i bagersproget, så drejer effektive algoritmer sig om at bruge mindst mulig tid og bøvl på at hente de enkelte redskaber og ingredienser, der skal bruges til brødet.

Det tager f.eks. lang tid at bage en kage, hvis man kun har plads til mel og gær på bordet, og skal flyve til f.eks. Australien efter en ske, hver gang man skal bruge den, til USA efter vand, til Kina efter sukker, til Brasilien efter bagepapir og Hawaii efter æg.

Det lyder vildt, men det er faktisk sådan, mange computerprogrammer fungerer, selv i vore dage. Når de henter data på harddisken, kan det sammenlignes med at hente mælk på den anden side af kloden, for det tager en million gange mere tid at hente data på en harddisk end at hente dem i cachen eller RAM'en, som vi her passende kan sammenligne med køkkenbordet.

Modsat RAM'en er harddisken nemlig mekanisk, den består af flere roterende skiver, hvorpå dataene ligger spredt og skal findes af en bevægelig arm.

Derfor gælder det om at minimere antallet af besøg på harddisken. Det kan gøres ved at sætte mere RAM i computeren, men der er grænser for, hvor mange RAM-blokke computeren kan håndtere.

Det næstbedste er derfor effektive input/output (I/O) algoritmer, som sørger for, at computeren henter flest muligt af de data, programmet skal bruge, hver gang den besøger harddisken. Man skulle måske tro, at computerne gjorde det i forvejen, men det er langt fra tilfældet.

Det skyldes ifølge Lars Arge, at algoritmemagere og programmører hidtil er gået ud fra en forsimplet model, hvorefter processoren bruger 1 tidsenhed på at hente et hvilket som helst vilkårligt data i hukommelsen. Med andre ord går modellen ud fra, at computeren har uendelig hukommelse, men det har den kun, hvis datamængden ikke overstiger hukommelsen i den enkelte maskine. Er der tale om massive data, passer modellen ikke.

God til at sortere

Effektive I/O algoritmer til massive data er Lars Arges speciale og ét af Madalgos fire fokusområder.

»Det gælder om at udnytte, at computeren ikke kun henter de specifikke data, man beder den om, men en blok data på 8, 16, 32 eller 64 Kbytes ad gangen. Dermed vil der typisk komme nogle data med, som man ikke skal bruge, men som blot fylder op – samtidig med, at der skal bruges tid på at hente de øvrige data, der skal bruges.

Hvis man fra begyndelsen sorterer dataene på harddisken i en smart rækkefølge, vil computeren uvægerligt hente blokke med data, som alle skal bruges på et tidspunkt i regneprocessen, og meget tid vil være sparet. Og vi er rigtig gode til at sortere, så dataene ligger rigtigt,« siger Lars Arge.

Det beviste han, da hans forskergruppe på Duke University i North Carolina for et par år siden skulle lave en simulation af vand-*flowet* på en elektronisk terrænmodel over et område på 64 mio. hektar i de amerikanske Appalache-bjerge.

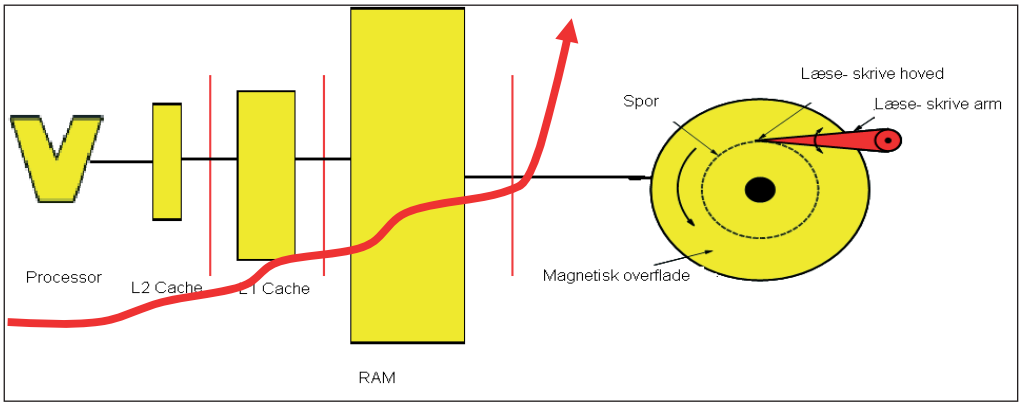
»Det handlede om at analysere dataene i den rigtige rækkefølge. Man inddeler i felter og sorterer efter højde først, derefter forsyner man hvert felt med data fra nabofelternes højde. Dermed nidobler man sådan set datamængden, men til gengæld kan man nu nemt sørge for, at alle de data, man henter i hver blok, kan bruges i det øjeblik, de hentes. Det nedbragte computertiden for en simulation fra to uger til tre timer – på den samme maskine,« forklarer Lars Arge.

For at vende tilbage til bageranalogien svarer det til at sørge for, at de ingredienser og redskaber, som der ikke er plads til på bordet, på forhånd er placeret ved siden af hinanden på den samme hylde i den samme butik.

Streaming data

I/O algoritmer kan imidlertid ikke benyttes på alle store datamængder. Nogle data bliver aldrig lagret, eller ikke lagret længe nok til, at man kan hente dem flere gange (f.eks. fra overvågningskameraer eller sensorer) – eller også er der tale om så enorme datamængder, at man kun har råd til at læse dem én gang.

Så bruger man streaming data algoritmer, et andet af Madalgos fokusområder, som tager udgangspunkt i at læse dataene



Den røde kurve viser arbejdstiden for bearbejdning af data, efterhånden som dataene skal hentes på langsommere medier – som f.eks. harddisken.

i den rækkefølge, de kommer fra en sensor eller ligger på en harddisk.

»Det bruger man f.eks. til overvågning af trafik på et netværk, hvor det kun er i nuet, man skal bruge dataene. Da handler det om at bruge så lidt tid og plads som muligt på hvert stykke data. Det kan være et telenetværk, der bruger algoritmer til at følge med i, om der opstår fejl eller hvor der skal sættes ind med ekstra kapacitet,« siger Lars Arge.

I stedet for at tælle alle dataene, der strømmer forbi, kan man f.eks. lave algoritmer, der med meget lidt plads og meget stor sandsynlighed kan beregne, hvilke datasæt der oftest forekommer i datastrømmen.

Aktuel og praktisk anvendelse

Et tredje fokusområde på Madalgo er algoritmer, der populært sagt er ligeglade med, hvilke hukommelsestyper, dataene hentes fra.

Her handler det om at skrive og læse så lidt som muligt på alle niveauer, både i cachen, RAM'en, harddisken og i princippet alle former for hukommelse.

Målet er at nå så langt, at den samme algoritme vil kunne bruges på alle platforme – lige fra supercomputeren til mobiltelefonen.

»Det er det mindst udviklede og mest teoretiske område af de fire, som centret beskæftiger sig med. Der er dog sket nogle gennembrud på det sidste, idet man f.eks. har vist, hvordan man sorterer optimalt på samtlige niveauer på en vilkårlig maskine – i hvert fald i teorien,« siger Lars Arge.

Et fjerde og sidste fokusområde på Madalgo kalder han *algorithm engineering*. Det er forskning i, hvordan man bruger de øvrige teorier i praksis.

»Vi vil forsøge at implementere praktiske algoritmer og lave eksperimenter med dem, for derefter at bruge resultaterne til at videreudvikle teorierne. Der er ikke så langt fra grundforskning til praktisk anvendelse i algoritmer for massive data, for motivationen er meget aktuel: datamængderne stiger ekspontielt,« konstaterer professoren. ■

Informations-Tsunamien

Yottabyte (1.000.000.000.000.000.000.000.000 bytes eller 10^{24} bytes)
Zettabyte (1.000.000.000.000.000.000.000 bytes eller 10^{21} bytes)
161 exabytes: Al digital information i 2006

5 exabytes: Alle ord der er sagt i menneskehedens historie
2 exabytes: Al information genereret i verden frem til 2002
Exabyte (1.000.000.000.000.000.000 bytes eller 10^{18} bytes)

200 petabytes: Alt trykt materiale
Petabyte (1.000.000.000.000.000 bytes eller 10^{15} bytes)
10 terabytes: Hele den trykte del af U.S. Library of Congress
Terabyte (1.000.000.000.000 bytes eller 10^{12} bytes)

50 gigabytes: En etagefuld bøger
5-7 gigabytes: 1 spillefilm på DVD
Gigabyte (1.000.000.000 bytes eller 10^{9} bytes)

700 megabytes: En CD-ROM
100 megabytes: 1 hyldemeter bøger
10 megabytes: Et minut lyd i Hi-Fi
5 megabytes: Shakespeares samlede værker eller et digitalt foto.
1 megabyte: En lille roman eller en 3,5 tommes floppy disk
Megabyte (1.000.000 bytes eller 10^{6} bytes)

100 kilobytes: Et digitalt foto i lav opløsning
2 kilobytes: En maskinskrevet side
Kilobyte (1.000 bytes eller 10^{3} bytes)
8 bytes: Et ord
1 byte: Et bogstav eller tegn
Byte (8 bits)
Bit (Et binært tal – enten 0 eller 1) *(Tallene beregnet med SI-præfix.)*

Om forfatteren

Peter F. Gammelby er journalist ved Alexandra Instituttet A/S
Tlf.: 8942 5770
E-mail: gammelby@alexandra.dk

Kontakt til professor Lars Arge: large@daimi.au.dk

Videre læsning:
How much Information? University of California at Berkeley, 2001

IDC-hvidbogen: The Expanding Digital Universe, marts 2007.

www.madalgo.au.dk

www.alexandra.dk?