

Systemarkitektur i forandring

Forandring, forandring og atter forandring. Neida, det er fortsatt korrekt som vi ofte har påpekt, at forandringene ikke skjer så fort som leverandører og enkelte andre vil ha det til. Stormaskiner lever i beste velgående, PC-arkitekturen er i hovedsak den samme som for 10 og 15 år siden, og gigahertzene har ikke bidratt til stort annet enn å holde tritt med stadig mer ineffektiv programvare.

Forandring, evolusjon og revolusjon

Forandringene kommer, om ikke så brått. Mens det er lett å finne eksempler på gammel teknologi og arkitektur som lever i beste velgående, er det naturligvis like lett å finne det motsatte. Evolusjon og utvikling hører til dagens orden, mens mer betydningsfulle forandringer er sjeldnere kost – og har vært bemerkelsesverdig fraværende de siste årene, til tross for at de fleste synes å tro det motsatte. Men fra tid til annen tvinger det seg frem vesentlige endringer, brudd med det tradisjonelle – enten fordi det gamle harmonerer dårlig med behovene, eller fordi nyskapninger bryter radikalt med tradisjonell tankegang og åpner nye og spennende muligheter.

Etter talløse år med beskjedne endringer utover de årlige ytelsesjusteringene, er turen kommet til laveste nivå i våre systemarkitekturer. I kjølvannet av 'Webifiseringen' av brukerprogrammer og voksende standardisering av ytelseskomponenter (*blades*), står vi foran en liten revolusjon på arkitektursiden. Hvor stor den blir, kommer an på en rekke faktorer, ikke minst i hvilken grad system-, hardware- og programvareleverandører makter å bremse utviklingen. Som vanlig er i forbindelse med vesentlige brudd i en evolusjon, vil aktørene som kontrollerer markedet, gjøre sitt ytterste for å motarbeide forandringen, fordi den representerer en trussel mot deres etablerte posisjon. Dette er ikke verken overraskende eller kritikkverdig, men like naturlig som at vinter blir vår. Først dersom aktørene tar uetiske eller ulovlige midler i bruk for å bevare status quo, blir imidlertid kritikk og sterkere lut på sin plass.

I denne artikkelen retter vi søkelyset mot vesentlige endringer som avtegner seg på teknologi-horisonten med hensyn til systemarkitekturer på laveste nivå. En oppfølgingsartikkel i neste utgave diskuterer de sterkeste drivkreftene bak denne revolusjonen, som ikke kan unngå å få betydelige konsekvenser på en rekke nivåer i IT-næringskjeden: Web-tjenester og portaler. Utgangspunktet for alle disse forandringene er Web-teknologien, hvis transaksjoner og kommunikasjonsbehov er helt annerledes enn tradisjonelle løsninger, inklusive klient/tjener-systemer.

Systemarkitekturer i forandring

Systemarkitekturer - hva er nå det? Som tilfellet ofte er – det kommer an på øynene som ser og sammenhengen. Derfor skal vi forsikre oss mot misforståelser: Et 'system' er i denne forbindelse fysiske datamasjiner, *hardware* som skal realisere våre løsninger på en optimal måte. Med 'optimal' tenker vi først og fremst på ytelse i forhold til kostnader, herunder forbruk av energi, plass og andre ressurser, samt pålitelighet.

At *hardware*-arkitektur på lavt nivå kun er ett av en rekke elementer i ligningen som til sammen gir en optimal løsning, gjør ikke fokuseringen mindre interessant. Prosessorene er selve fundamentet for systemene, uansett hvor godt gjemt de måtte være. En lite egnet prosessorarkitektur kan aldri gi et kosteffektivt total-system.

Antikvarisk hardware?

Det er ingen grunn til å underslå at *hardware* i sin alminnelighet får stadig mindre fokus fra kunder og brukere. Det er tjenester, ikke hvordan de blir levert, som er i fokus. Men noen skal ta ansvaret for levering av tjenestene – drive infrastrukturen og etablere det tekniske grunnlaget. For disse miljøene er arkitekturen ikke bare viktig, men i mange tilfeller forskjellen mellom pluss og minus på resultatsiden. Og til tross for *outsourcing* og ASP-bølger, leverer organisasjoner flest sin egen datakraft, og hører derfor til denne gruppen.

Dessuten kan vi observere at utviklingen på løsningssiden, der Web-tjenester og portaler overtar en jevnt voksende del av totalen, så langt ikke i nevneverdig grad har påvirket utviklingen av verken *hardware* eller operativsystemer. Dette spriket mellom behov og tilgjengelige verktøy er verken unikt eller av ny dato. For eksempel har de fleste av oss observert at operativsystemene ikke holder følge med *hardware*-utviklingen. Spesielt Microsoft har vist en foruroligende sendrektighet i forbindelse med støtte av først 32-bits, så 64-bits arkitekturer.

Nå viser det seg altså at det samme kan sies om *hardware* – fra prosessorer og opp til hele systemer: De konstrueres fortsatt som om verden ikke har forandret seg siden 1995, med fokus på flyttallsberegninger, MIPS og ulike benketester hvis relevans gikk ut på dato for snart 10 år siden. Vel har vi fått både nye operativsystemer, *blade*-tjenere, nye prosessorer, lagringssystemer og så videre, men er de tilpasset dagens belastningsbilde og tilsvarende behov, for ikke å snakke om de behov vi ser for den overskuelige fremtid? Praktisk talt aldri.

Denne underlige mangel på synkronisering i et marked de fleste oppfatter som både dynamisk og 'i forkant', har først og fremst sin rot i divergerende interesser i ulike deler av næringskjeden. Microsoft blir hyppig beskyldt for å bremse utviklingen fordi den ikke passer selskaps lommebok eller verdensanskuelse, og er i godt selskap: Slik bremsing – overlatt eller på grunn av innsiktsløshet – er regelen snarere enn unntaket i bransjen.

To tett beslektede forhold er spesielt interessante i den sammenheng: Hvilke krav stiller moderne løsningsarkitekturer til den underliggende *hardware*, og hvordan tilfredsstilles disse kravene av tilgjengelig jern?

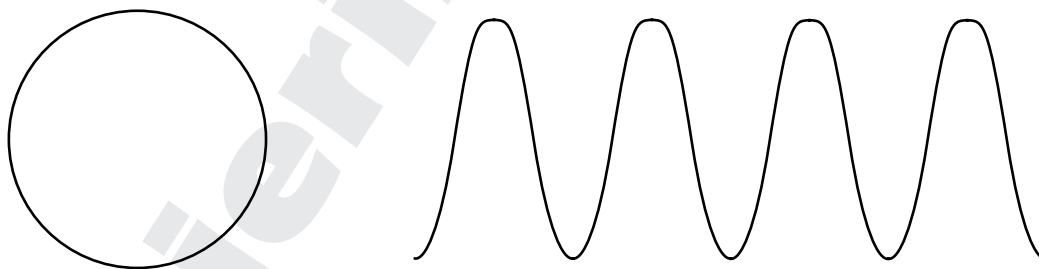
'Jern' og 'maskiner' er det samme, og består av en lang rekke komponenter hvorav prosessoren (entall eller flertall) udiskutabelt er den viktigste. Derfor er det naturlig å begynne med et overordnet perspektiv på nettopp prosessorfronten, der Gigahertz er blitt tidens målestokk til tross for at klokkefrekvensen sier fint lite om den egentlige ytelsen.

Teknologiske sirkler

Prossessorer, hastigheter og utvikling

Når utviklingen tilsynelatende går i sirkel, hvilket vi kan observere på en rekke områder fra tid til annen, er det som regel vår synsvinkel som er gal. En spiral ser alltid ut som en sirkel rett forfra eller bakfra. Tynne kontra tykke klienter er et nærliggende eksempel: Vi har gått fra 'dumme terminaler' – en historisk betegnelse på datidens tynne klienter, via PC-kraftverk på alles skrivebord – og er på full fart tilbake til tynnere og enklere varianter. Forskjellen mellom 70- og 80-tallets terminaler på den ene siden og våre dagers Windows-terminaler – nedbestykkede Pcer og SunRay'er – på den andre, er liten i en prinsippkisse, men stor både teknisk og brukermessig.

Samme observasjon kan gjøres for stormaskiner. Både fysisk størrelse, kostnader, anvendelser og krav til driftsapparat er forbausende like mellom 70-tallets stormaskiner og dagens store tjenere fra IBM, SGI, Sun og HP. Et tredje eksempel er 80-tallets løsninger kontra dagens – der målsettingene ofte er omtrent de samme, mens kompleksitet, omfang, samspill og tilgjengelighet er av en annen størrelsesorden. Vi er ikke tilbake til utgangspunktet – det er utviklingen som har gått en full omdreining – langs en spiral.



Figur 1 En spiral har dramatisk forskjellig karakteristikk avhengig av synsvinkel. Sett fra enden er den en sirkel, mens sett fra siden observerer vi en sinuskurve med jevne svinginger. Teknologiu utviklingen har mange fellestrekk med en spiral. Fra tid til annen kan det se ut som om vi går i sirkel, mens vi i realiteten har fullført en hel runde langs en spiral. Fremdrift og utslag kan variere, og tilsvarende observasjoner kan gjøres for svært mange andre områder i tillegg til teknologi.

Dermed kommer det knapt som en overraskelse at vi ser omtrent det samme forholdet når vi retter søkelyset mot prosessorsiden. 80-tallets RISC-teknologi var en reaksjon på voksende kompleksitet i prosessorne, som i løpet av 70-tallet hadde gått fra det helt enkle til sammensatte og ressurskrevende instruksjonssett – etter datidens målestokk: Over 100 instruksjoner i flere arkitekturer – for flyttallsoperasjoner,

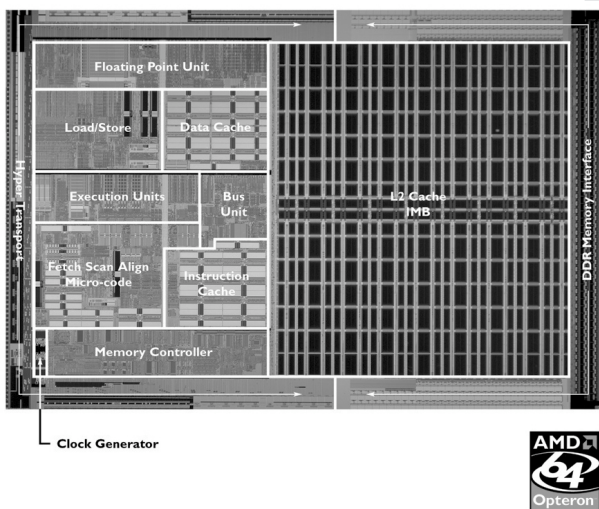
strengbehandling og så videre. Argumentet var at alle disse instruksjonene gjorde hele arkitekturen mindre effektiv, og at de komplekse operasjonene kunne gjøres raskere i programvare. En arkitektur med et fåtall enkle og særdeles effektive instruksjoner som kombineres til den grad av kompleksitet som trengs – i programmene i stedet for i *hardware*. En viktig faktor i den forbindelse var at de kompliserte instruksjonene ble relativt sjelden brukt. Fjerningen av dem bidro til en betydelig akselerasjon av de enkle og mye brukte instruksjonene – med innlysende konsekvenser for total-ytelsen.

Fra CISC til RISC – og tilbake

Et titalls RISC-baserte arkitekturer med utgangspunkt i forskning fra Berkeley og andre amerikanske universiteter dukket opp – en generasjon av enklere og raskere prosessorer. Et tosifret antall nye aktører kom på banen – alle med sine egne RISC-arkitekturer, men uten tilstrekkelige ressurser til å overleve i det lange løp. Selv Intel, som forble CISC-tankegangens bastion både markedsføringsmessig og praktisk, garderte seg ved å utvikle en serie RISC-prosessorer¹ og tok erfaringene fra disse med over til både x86- og andre områder.

RISC-tankegangen var forankret i et problem: Kompliserte instruksjoner tok lang tid å utføre. Over tid ble imidlertid problemet borte. Den underliggende chip-teknologien ble mer avansert, barrierer ble brutt og nye muligheter åpnet seg. Siden midten av 90-tallet har derfor de overlevende RISC-arkitekturerne – Alpha, PA-RISC, SPARC, PPC og et fåtall andre – blitt vanskelige å distingvere fra sine CISC-konkurrenter, i hovedsak representert ved Intels x86-serie. At leverandørene fortsetter å kalle sine arkitekturer for RISC eller CISC, har primært markedsføringsmessige årsaker. Alt er CISC, og konfrontert med de faktiske forhold er det ingen som benekter dette.

Typisk for prosessorutviklingen de siste årene er krympende transistorer, økende kompleksitet og sterkt voksende klokkefrekvenser – med tilhørende høyere ytelse. Den reelle ytelsen har imidlertid ikke vokst proporsjonalt med økningen i klokkefrekvens. Vår 2 GHz PC går ikke 4 ganger raskere enn vår forrige, 500 MHz maskin.² 'Ytelse måles ikke i Megahertz eller Gigahertz' får vi vite. Det har mange av oss vært klar over lenge, men hvorfor fortsetter leverandørene å fokusere på denne meningsløse måleenheten? Svaret er innlysende – og dumt: 'Fordi det virker'. Mer meningsfylt blir det imidlertid ikke av den grunn. Selv Intel-



Figur 2 AMDs ferske Opteron-prosessor har en tradisjonell arkitektur med hensyn til hvordan instruksjoner behandles og utføres, og en ny I/O-arkitektur. Den består av over 40 millioner transistorer, hvorav de fleste er hukommelse (L1 eller L2 cache).

¹ Best kjent av disse er i960-serien.

² Ifølge benketester fra PC Magazine er den praktiske ytelsesforbedringen fra Intel Celeron 1,2 GHz til Intel Pentium 4 2,2 GHz på ca. 20%.

grunnlegger Gordon Moore's berømte lov blir misrepresentert ofte nok til at de fleste i dag tror den har med ytelse å gjøre. Det har den ikke. Moore's lov snakker om transistor tetthet i forhold til tid,³ verken mer eller mindre.

Men tilbake til prosessorer: Utviklingen har åpenbart gått en hel runde fra midten av 80-tallet til siste halvdel av 90-tallet. Komplekse instruksjonssett er tilbake – og er grunnlag for prosessorer med en kapasitet og kompleksitet vi i beste fall drømte om for 5-6 år siden. Og utviklingen tar ingen pause. Mulighetene forandrer seg, flaskehalsene og dermed utfordringene flytter seg.

På laveste nivå har den voldsomme økningen i klokkehastighet skapt en interessant situasjon: Mens prosessorene kan behandle data dramatisk mye raskere år om annet, er hastigheten med hvilken data kan hentes fra hukommelsen omtrent uforandret. Derfor opptar mellomlagre på flere nivåer (*cache*) det meste av plassen på dagens prosessorbrikker. Disse mellomlagrene kan ha mange navn, men er ikke noe annet enn turboladet hukommelse. Videre inneholder brikkene komplisert logikk som skal forutse programmenes gang (*branch prediction*), slik at avgreninger – som alle programmer er fulle av – ikke forårsaker total stans i utførelsen.

At brikkene må inneholde så mye hukommelse for å holde oppe rimelig hastighet, er åpenbart et problem som bare blir verre – inntil noen kommer opp med en idé som bringer oss til neste nivå: Med dagens utvikling blir prosessorene om mindre enn to år for rene hukommelsesbrikker å regne – hvilket vitterligen fungerer, men som samtidig representerer en enorm sløsing med ressurser sett fra alle synsvinkler.

Forandring i sikte?

Leverandørene – Intel, AMD, Sun, IBM, HP og flere – har naturligvis brukt enorme ressurser på denne utfordringen i lang tid, uten å finne noen tilfredsstillende løsning. Sannsynligheten er stor for at de leter på feil sted – eller i gal retning – siden resultatene er så totalt fraværende. Om løsningen uteblir, dukker det imidlertid opp spennende ideer og initiativer fra tid til annen. For eksempel har AMDs nye Opteron 32/64-bits arkitektur (figur 2) en ny I/O-arkitektur som kan vise seg å være av vesentlig betydning for ytelsen. Likeledes presenterte Sun Microsystems nylig et høyst interessant konsept som på mange måter representerer en retningsendring tilbake mot RISC-tankegangen.

Som ofte er tilfellet når avgreninger fra den etablerte utviklingsretningen forekommer, er det en nykommer som står bak. Selskapet Afara Websystems ble etablert av avhoppere fra Sun som mente videreutviklingen av SPARC-prosessorer – og prosessorarkitekturer i sin alminnelighet – gikk i gal retning. De angrep utfordringen med utgangspunkt i to grunnleggende forutsetninger eller erkjennelser:

³ Moores lov: "The number of transistors on a chip doubles every 18 months."

Den første er at generelle prosessorer ikke er velegnete til alle slags oppgaver, snarere tvert imot. Tankegangen er den samme som vi har påpekt ved en rekke anledninger i Mellvik-Rapporten: At universal-systemer og -løsninger er vel og bra, men aldri blir optimale i noen sammenheng.

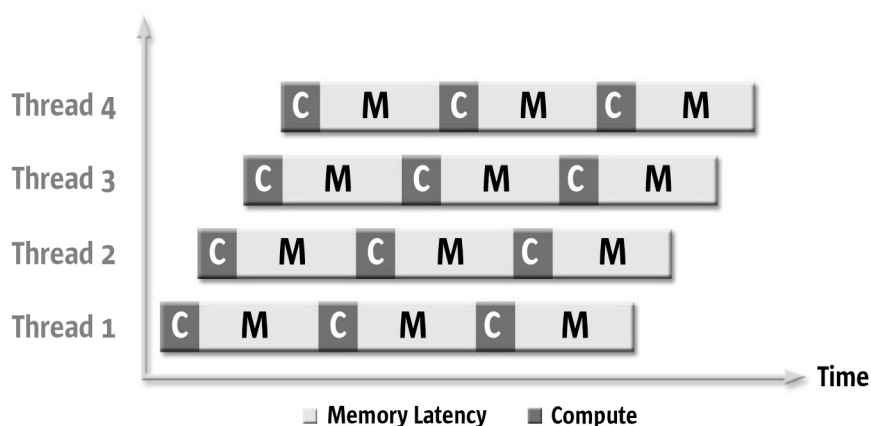
Slik er det naturligvis også med prosessorer. Dagens typiske mikroprosessor – IA32, IA64, Alpha, UltraSPARC, PPC og så videre – kan brukes til det meste, men er med enkelte unntak ikke optimale for noe. Derfor finner vi dem i like generelle systemer, og sjelden i spesialisert utstyr som mobiltelefoner, PDAer og industrielle produkter. Verden er full av oppgaver som fordrer skreddersøm og spesialtilpassede løsninger – etter samme tankegang som vi bruker hammer til spiker, skrujern til skruer og ikke kniv til begge deler.

Overgangen til Web-baserte løsninger og tjenester er, som vi allerede har nevnt, en viktig bakenforliggende drivkraft i dette bildet. Og her er vi ved ovenfor nevnte Afara Websystems forutsetning nummer 2: Dagens og morgendagens typiske løsninger karakteriseres av korte og selvstendige I/O-transaksjoner. Slike oppgaver har liten nytte av avanserte flyttallsalgoritmer og trippelindekserte sidetabeller for virtuell hukommelse. Faktum er at en rekke av de mest kompliserte elementene i en generell prosessor kan forsakes med utelukkende positive konsekvenser for ytelsen. Ved å foreta en slik forenkling, kan prosessorenes kompleksitet, størrelse, strømforbruk og dermed pris reduseres vesentlig.

Alternativt kan forenklingen benyttes til å gi plass til flere selvstendige prosessorer på samme brikke – en idé som allerede har vært testet med hell i ulike andre sammenhenger.

Afaras idé stoppet imidlertid ikke der. Selskapet påpekte at slik belastningsbildet for typiske Webtjenere ser ut, vil det være en fordel å flytte flertrådsbehandlingen som er karakteristisk for dagens operativsystemer, inn i selve prosessoren. På den måten kan forsinkelsen som alltid påløper når data må hentes fra hukommelsen, utnyttes til å kjøre andre aktive 'tråder'. Konsept og tankegang er iden-

Chip Multithreading (CMT)



Figur 3

Flertrådsteknologi har i voksende grad funnet veien til operativsystemer siden slutten av 80-tallet, med godt dokumenterte, positive effekter. At mekanismen nå flyttes ned i selve prosessoren er en rimelig utvikling – som nødvendigvis må gå på bekostning av noe. Og hva er mer rimelig enn å fjerne komplekse elementer som sjelden brukes i de fleste sammenhenger? Illustrasjonen viser hvordan flere aktive tråder kan holde en rask prosessor i gang kontinuerlig – mens de ulike trådene venter på data fra hukommelsen. Illustrasjonen er laget av Sun Microsystems, som ikke er alene om å tenke i disse baner.

tisk med velkjent, operativsystem-basert flertrådsteknologi, og argumentet for å benytte metoden er akkurat den samme.

Å flytte mekanismer og funksjonalitet fra operativsystem til prosessor på denne måten, er en velkjent progresjon som historien har tallrike eksempler på. Likeledes er både eksemplene som brukes og simuleringer som er gjort, indikative på at dette er en god idé.

Afaras ideer og selskapets utvikling av SPARC-basert teknologi for å realisere ideene, vakte tilstrekkelig interesse hos Sun Microsystems til at de like godt kjøpte selskapet i fjor sommer. Siden har utviklingen fortsatt og de første resultatene ble annonsert tidligere i høst. Konseptet kalles for CMT, *Chip Multithreading* (figur 3) og er grunnlag for en ny prosessorserie som skal introduseres på vårparten neste år. Det første produktet ('Gemini') har to regne-elementer (prosessorer) som hver kan ha flere aktive tråder. Skisser av neste generasjon ('Niagara') med 6 regne-elementer ble presentert samtidig.

Sun Microsystems er ikke alene om å tenke i disse baner. I forbindelse med lanseringen av den PPC-baserte P5-arkitekturen nylig, presenterte IBM tanker i samme retning. Her er definitivt noe i gjære, en utvikling som i forhold til markedets behov, er overmoden.

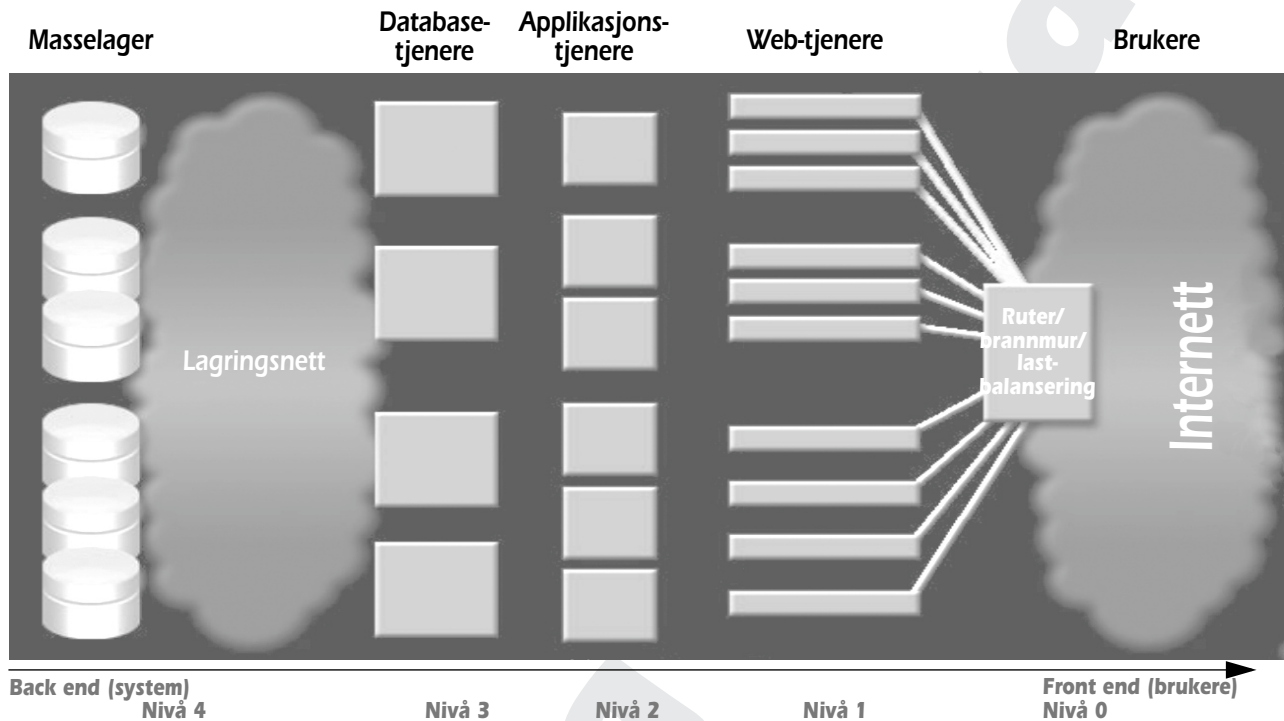
Fra hardware til system

Hvorvidt Suns, IBMs og AMDs ideer og den praktiske implementasjonen av dem tåler markedets test, gjenstår å se. Imidlertid forteller deres eksistens at vi er på vei over i en ny fase også på laveste hardware-nivå. Ikke bare blir *blades* den viktigste hovedkomponenten i fremtidens datasystemer, men prosessorene som blir å finne på disse '*bladene*' vil i voksende grad være oppgavetilpassede i stedet for generelle. Virtualiserings-trenden både aksentuerer og legitimerer denne utviklingen gjennom å gjemme de underliggende detaljene fra operativsystemer og løsninger – et forhold vi blant annet diskuterte i artikkelen om lagringssystemer i Mellvik-Rapporten nr. 101.

Prosesor-nivå er ikke alene om å være utsatt for spennende forandringer i disse dager. Om vi flytter oss 3 hakk oppover, til nettverket, ser vi konturene av en tilsvarende revolusjon i miniatyr. Katalysator er tilgjengeligheten og de lave prisene på Gigabit Ethernet teknologi, et forhold vi også var inne på i forrige utgave (Mellvik-Rapporten nr. 108 side 26). Standardkomponenter og standardisert transport-teknologi – i hovedsak basert på IP, kombinert med nye protokoller som iSCSI, provoserer frem nytenking når totalsystemer skal settes sammen. For eksempel blir nettverket raskere enn mange interne system-bus'er, slik at det blir både billigere og mer effektivt å flytte både masselager og andre komponenter ut fra databehandlingsenhetene.

Likeledes, og igjen som en følge av det vi kan kalle 'Webifiseringen', blir det optimalt å fordele arbeidsbelastningen annerledes enn tidligere, typisk med rene transaksjonsprosessorer i forkant, applikasjonsprosessorer i midten og database- eller lagrings-prosessorer bakerst (se figur 4). Hypereffektive, svitsjede lokalnett basert på Gigabit

Ethernet eller 10 Gigabit Ethernet bidrar til at en slik oppsplitting ikke bare blir mulig, men sågar optimal – spesielt når den kan kombineres med spesialiserte prosessorer som vi diskuterte ovenfor.



Figur 4 Den tiltagende bruken av nettlesere og HTTP som rammeverk for brukertjenester og -løsninger, legger forholdene til rette for en vesentlig ombygging av systemarkitekturen. Å etablere nivåer som vist på figuren, gir både fleksibilitet og effektivisering, samtidig med at det blir mulig å etablere skreddersøm for de ulike oppgavene – i hardware såvel som i programvare. En rekke trender i tiden understøtter en slik arkitektur-utvikling, for eksempel større grad av virtualisering og kompakte BLADE-løsninger.

Diskløse tjenere?

I kjølvannet av det voldsomme prisfallet på halvlederbasert hukommelse (*Flash-memory* og alminnelig DRAM/SDRAM) de siste årene, aner vi en annen arkitektur-messig forandring: Behovet for tradisjonelt masselager reduseres og blir i enkelte tilfeller borte. Forutsigbare oppgaver og tilgang på masselager lenger bak i systemet gjør det mulig å forenkle både 1. (Web-transaksjoner) og etterhvert 2. (applikasjons-transaksjoner) nivå i hierarkiet.

Denne utviklingen, som vi skal komme tilbake til i neste utgave, gir et ytterligere bidrag til forenklingsprosessen, og ikke minst til fleksibiliteten. Å omallokere ressurser til andre oppgaver i takt med behovet blir trivielt i stedet for krevende, og kan gjøres på sekunder i stedet for timer.

Neste utgave

I neste utgave fortsetter vi fokuseringen på system-arkitektur med en gjennomgang av drivkreftene bak og konsekvensene av de forandringene vi nå har gjennomgått. Som vi allerede har nevnt, er 'Webifisering' en nøkkel i den forbindelse. Her står overforbrukte termer som

'portaler' og 'Web-tjenester' sentralt. I oppfølgings-artikkelen spør vi både hva disse begrepene inneholder anno 2003, hvilke gevinster de har å by på, utfordringene i deres kjølvann, når forandringene vil gi synlige resultater – og ikke minst: Sammenhengen mellom utviklingen på tjeneste- og løsningsnivå med de systemarkitekturerne vi har diskutert i denne artikkelen. ■