

Windows Services for Unix 3.0

Forholdet mellom Microsofts operativsystemer og Unix opptar mange og har vært diskutert ved tidligere anledninger i Mellvik-Rapporten. Spesielt aktuell i forbindelse med SFU er artikkelen "NT+UNIX: Odde frender i pragmatisk marked" i nr. 32 (september 1996).

"Mye vil ha mer ..." heter det i et velformulert ordtak, og suksess avler appetitt – ikke minst i markedsmessig forstand. Uansett hvilket segment vi tar for oss, er en markedsandel på et 2-sifret antall prosent for en suksess å regne. Når den passerer 50% er dominans det riktige ordet, og 80-90% betraktes i de fleste sammenhenger som de facto monopol.

Pågående monopolrettssaker i USA handler ikke om hvorvidt Microsoft har et *de facto* monopol, men om monopolet også eksisterer i juridisk forstand og dessuten misbrukes. Selskapets monopol gjelder for det vi kaller 'bordflaten' – brukermaskiner. På tjenersiden har vi ikke bare reelle valgmuligheter – vi observerer sågar at Microsoft i en rekke segmenter er praktisk talt fraværende. Mens dette forholdet er naturlig i og med at samme verktøy umulig kan være like godt egnet til alle slags oppgaver, representerer disse segmentene også store fristelser for en aktør som til de grader har klart å utradere konkurransen på andre områder.

Kompatibilitet, marked og historie

SFU – *Services for Unix* – er et tilleggs-produkt for Microsofts tjeneroperativsystemer (NT, W2k) som skal gjøre systemene Unix-kompatible, hvilket i praksis også betyr Linux-kompatible.¹ I utgangspunktet er dette en like stor selvmotsigelse som å gjøre Unix Windows-kompatible: Fundamentalt ulike systemer med helt forskjellig arkitektur og med sprikende angrepsvinkler på grunnleggende problemstillinger. Ulikhetene er like ideologiske som de er praktiske: Det var et vesentlig poeng for David Cutler, mannen bak Windows NT-arkitekturen, og for Bill Gates, å vise verden at tiden er ute for 'gammeldagse' operativsystemer som Unix. Vi befant oss i første halvdel av PC-epoken – tidlig på 90-tallet, der selvstendighet og uavhengighet – én bruker per maskin – var *credo*. Windows NT og etterfølgerne er naturlig nok preget av dette, og er i sin natur énbruker-systemer. *"Even Windows NT, although it is a multi-user system, has not provided for multiple Windows sessions simultaneously"*, påpeker Microsoft i SFU-dokumentasjonen.²

Best i verden

Å fokusere så ensidig på å betjene én bruker viste seg imidlertid å være en tabbe. Det kan være innlysende i dag, men i lys av suksessen med Windows 3.x er Microsofts valg forståelige, om aldri så historieløse. Etterhvert som 90-tallet skred frem, ble flerbruker-egenskaper ikke bare viktige, men nødvendige, og først NT, siden W2k fikk sine egne flerbruker-utvidelser (*Terminal Server Edition*, TSE).³

1 I denne artikkelen grupperer vi for enkelhets skyld ulike varianter av Unix og Linux sammen under fellesbetegnelsen 'Unix'.

2 Dokumentet heter "Interix in a Multi User TSE Environment".

3 Se Mellvik-Rapporten nr 54.

Utvidelsene tok vare på den primære og for Microsoft viktigste utfordringen knyttet til markedets reversering mot sentraliserte systemer: Å hindre en masse-desertering fra Windows-plattformen. Noe grunnlag for videre vekst i markedsandeler på bekostning av etablerte og fortsatt sterke Unix-aktører, representerte utvidelsene imidlertid ikke. Her måtte det annen lut til, kort og godt den samme medisin som konkurrentene hadde brukt i mange år, men som Microsoft inntil 1995 tok sterk avstand fra: Kompatibilitet og emulering. Holdningen var at 'vi er ledende og har ingen grunn til å forsøke å ligne på andre'.

Press fra markedet

Pipen fikk en annen lyd da det amerikanske forsvaret forlangte Unix-kompatibilitet for i det hele tatt å vurdere Windows-systemer i tjener-sammenheng. Windows NT ble i all hast utstyrt med en såkalt POSIX⁴-modul, en samling programmer og biblioteker som skulle tilfredsstille kompatibilitetskravet. Tillegget viste seg å være tilstrekkelig til å ikke bare komme i betraktning, men også å hale inn store ordrer. At programvaren kvalitetsmessig knapt kunne brukes til mer enn å gjennomføre enkle demonstrasjoner var kun unntaksvis et problem.

Behovet for reell kompatibilitet vokste imidlertid parallelt med appetitten på nye markedsandeler: Grunnleggende samspill med Unix var og er en forutsetning for å vurdere Windows som alternativ på tjenersiden for nettverks- og Internett-tjenester. God kildekodekompatibilitet måtte til for å muliggjøre flytting av applikasjoner og tjenester utviklet for Unix-miljø. Uten slik kompatibilitet blir portingskostnadene for store og gevinstene vanskelige å få øye på. Likeledes representerer utfordringen knyttet til drift en vesentlig kostnad som kan avhjelpes med god kompatibilitet den ene eller den andre veien.

Behovet for å bygge bro mellom Windows og Unix på såvel klient- som tjenersiden var tilstrekkelig stort til at en håndfull leverandører tidlig kom på banen. Produktene gjennomgikk kontinuerlige forbedringer og utvidelser i løpet av 90-årene, og gjorde et kvantesprang i forbindelse med at Windows NT kom på markedet. De opparbeidet seg tilstrekkelig anerkjennelse og posisjon i spesielle markedsnisjer til å bli synlige på radaren hos blant andre Microsoft [se fakta-rammen på neste side for detaljer] som – av årsaker vi diskuterte ovenfor – bestemte seg for å komme på banen for alvor.

Her var det liten grunn for Microsoft til å starte på bar bakke. I tråd med sunn fornuft – og egne tradisjoner – valgte selskapet å skaffe seg tilgang til grunnlagsteknologi og ekspertise fra den fremste aktøren i segmentet. Mortice-Kern Systems (senere omdøpt til MKS, Inc.) lisensierte viktige deler av sin teknologi til Microsoft for bruk i *Services For Unix* (SFU) versjon 2.0, som kom i 1999. Produktet representerte et kvantesprang i retning av reell kompatibilitet på kildekode-nivå for

⁴ POSIX er en spesifikasjon som definerer en rekke sider (programmeringsgrensesnitt, verktøy, biblioteker, tegnsett, dataformater og mer) med utgangspunkt i Unix-systemer. De fleste Unix-leverandører og Linux bekjenner seg til denne standarden.

Unix på Windows – produkter og aktører

Unix har siden 80-tallet vært spesielt populært i utviklings- og driftsmiljøer, blant annet på grunn av sitt rike tilfang av effektive verktøy. Da PCen og MS-DOS tok av for alvor rundt 1985, var derfor veien kort frem til de første samlingene av Unix-verktøy for DOS dukket opp. Disse var i utgangspunktet fritt tilgjengelige i kildekode (*Open Source*), og la grunnlaget for flere kommersielle og frie produkter som på 90-tallet tilbød et omfattende Unix-miljø for Windows. Flere av produktene, inklusive forløperen til Interix, ble omtalt i Mellvik-Rapporten nr. 32. De viktigste produktene i dagens marked er:

- **Cygwin** – en omfattende samling programmer, verktøy og biblioteker fra *Open Source* leverandøren Cygnus Research, som ble overtatt av RedHat Software i 2000. Hovedmålsettingen for produktet er å etablere et programmeringsmiljø med tilhørende verktøy for porting av Unix-applikasjoner til Win32-miljø. Produktet støtter alle Windows-versjoner siden W95, med unntak av CE. Support er tilgjengelig for betalende kunder. Siste utgave av Cygwin blir omtalt i vår Godbiter-spalte i september-utgaven av Mellvik-Rapporten (se baksiden). <http://cygwin.com/>
- **UWIN** – fra AT&T Research, hvor Unix trådte sine barnesko tidlig på 70-tallet. Pakken er fritt tilgjengelig for individuelle brukere, lisensieres for kommersiell bruk, forefinnes delvis i *Open Source*, og utgjør et særdeles komplett og effektivt utviklings- og driftsmiljø for NT og W2k (se omtale i Mellvik-Rapporten nr. 50). www.research.att.com/sw/tools/uwin/

Mens begge disse holder høy standard, er de verken så komplette eller stabile som SFU 3.0. Likeledes er graden av kompatibilitet gjennomgående noe mer beskjeden og hjelpemidlene for installasjon og drift mindre sofistikerte. Hvorvidt ulik tilgang på support spiller noen praktisk rolle, er et åpent spørsmål.

I den andre enden av skalaen finner vi VMWARE, som ikke emulerer et miljø, men en maskin-arkitektur – se egen ramme på side 8.

Andre produkter som angriper deler av 'kompatibilitetsproblematikken' er:

- **WinXs** – fra Software Online Limited, en samling Unix-programmer og verktøy portet til Windows. Et nyttig, men langt fra komplett tilskudd til et hvilket som helst Windows-system (*shareware*). www.softwareonline.org/winxs42.html
- **Omni** fra Xlink Technology – en samling kommersielle produkter sentrert rundt NFS, printserver, Xwindows og terminalemulering. www.xlink.com
- **MKS Toolkit** – fra MKS Software, tidligere Mortice-Kern Systems – et relativt komplett utviklingsmiljø for porting av applikasjoner fra Unix til Windows: Biblioteker, verktøy for utvikling, debugging og test samt et komplett Unix-miljø som på mange områder kan måle seg med SFU 3.0. MKS Toolkit er nå integrert med det tidligere konkurrerende produktet **NuTCracker** etter at MKS slo seg sammen med Datafocus tidligere i år. www.mkssoftware.com
- **Interix** – fra Interop Systems (tidligere Softway Systems), som er en hovedbestanddel i Microsofts nye SFU 3.0, og tar utgangspunkt i de samme *Open Source* verktøy og programmer som Linux og såvel frie som kommersielle Unix-varianter. www.interopsystems.com
- **Reflection** – fra WRQ Software (www.wrq.com) – en samling kommersielle 'interoperabilitets'-pakker med vekt på NFS, Xwindows og terminalemulering.
- **Exceed** og **NFS Maestro** fra Canadiske Hummingbird har mye til felles med produktene fra WRQ (NFS og X11 fokus) og har konkurrert i samme segment siden tidlig på 90-tallet. www.hummingbird.com

Videre finnes det en rekke enkeltstående verktøy som angriper ulike problemstillinger knyttet til integrasjon og samspill mellom Unix og Windows systemer, for eksempel synkronisering av passord, deling av filer og printere, og så videre.

Windows NT i forhold til den opprinnelige POSIX-modulen, og bød på reell konkurranse for de etablerte aktørene i nisjen, inklusive MKS.

Fremskrittet ga tydelig mersmak. Samtidig hadde SFU 2.0 gitt uvurderlige erfaringer med hensyn til hvor de store utfordringene lå og hva som skulle til for å ta steget fra delvis til bortimot full kompatibilitet. MKS' teknologi ble forkastet til fordel for Interix fra selskapet Interop Systems. Interix er – i likhet med Linux og en rekke av dagens Unix-systemer – i stor grad basert på *Open Source* programvare sentrert rundt Free Software Foundation og det såkalte GNU-konseptet.⁵ Resultatet er SFU 3.0, som ble tilgjengelig på markedet i slutten av juni måned.

⁵ Se "Open Source: Tilbake til virkeligheten" i Mellvik-Rapporten nr. 64, "Fri Programvare: Økologi for en doped IT-verden" i nr. 54 og <http://www.fsf.org>.

SFU 3.0 – 95%+ kompatibel

NIS – *Network Information Services*, nettverksbasert katalogtjeneste som har vært i bruk i Unix-miljøer siden slutten av 80-tallet.

PAM – *Pluggable Authentication Module*, modulær autentiseringsmekanisme som er utbredt i Unix-miljøer.

GPL – *GNU General Public License*, lisensbetingelser som benyttes for de fleste *Open Source* programmer og prosjekter, utarbeidet av **FSF**, *Free Software Foundation*. **GNU** er FSFs varemerke, og GPL betegnes ofte som 'copyleft' – som motvekt til 'copyright'.

I lys av sine forgjengere fra Microsoft og andre aktører – kommersielle og frie – er SFU 3.0 et imponerende stykke programvare. Emuleringen av Linux/Unix-miljø er bedre enn de fleste av oss trodde var mulig, spekteret av tjenester er bredt og effektiviteten er i de fleste tilfeller forbausende god. Vi har installert og anvendt SFU på flere systemer de siste tre månedene, og deler den oppfatning vi har fått fra andre testmiljøer: Forkledningen er god, hjelpemidlene er av høy kvalitet og driftsmessige forhold er vel tatt vare på.

Før vi ser nærmere på hva dette betyr i praksis, er det imidlertid viktig å minne om at vi snakker om kildekode- og bruks-kompatibilitet: Kompilerte programmer fra en eller annen Unix-plattform kan ikke kjøres direkte, slik tilfellet er motsatt vei – med emulatorer som Wine (Mellvik-Rapporten nr. 59 og 93), og avledede produkter som Lindows (se side 34). SFU får NT/W2k til å opptre som et Unix-system, kompatibelt med andre Unix-systemer med hensyn til administrasjon, brukergrensesnitt, standardverktøy, tjenester, kildekodekompatibilitet og biblioteker. I SFU 3.0 finner vi en rekke elementer som bringer samspillet – som klient, tjener eller begge deler – til et hittil ukjent nivå:

- ✓ Autentisering (NIS-tjenester som samspiller med NT Domain tjenester og Active Directory).
- ✓ Synkronisering av passord – PAM gir synkronisering mot autentiserings-infrastrukturen som er standard i mange Unix-miljøer.
- ✓ Tjeneste for 'mapping' av brukernavn – et tilsynelatende uløselig problem i og med at brukernavn representeres helt ulikt i Windows og Unix-miljøer.
- ✓ NFS-støtte – klient, tjener og *gateway*.
- ✓ Generelle nettverkstjenester – kjente pakker som *sendmail*, *postfix*, *apache*, *zope* osv. kan installeres og driftes som på Unix-systemer.
- ✓ Driftskompatibilitet – kjente driftsverktøy fra Unix-miljø er inkludert, og kommandofiler (*perl*, *shell* etc.) kan stort sett brukes uten videre, og andre kjente verktøy kan installeres uten spesielle tilpasninger.

Både for anvendelse og drift er det viktig at såvel fri som kommersiell programvare kan installeres og kompiles direkte, normalt uten noen form for tilpasning. At SFU 3.0 gjør dette mulig er i seg selv bemerkelsesverdig i og med at utgangspunktet – vertssystemet – er så forskjellig fra Unix med hensyn til arkitektur og egenskaper.

Windows + OPEN SOURCE = Sant

På alle nivåer i SFU skinner det igjennom at fokus har vært rettet mot maksimal likhet i stedet for minste felles multiplum, som har vært typisk for tidligere produkter i samme kategori. De fleste medfølgende verktøy er identiske med hva vi finner i et Linux-system, og prosessen for å installere nye pakker er den samme. Microsofts avstandtagen fra *Open Source* og GPL hører åpenbart hjemme i kategoriene retorikk og

VMWARE: Virtuelle maskiner til heder og verdighet igjen ...

Først IBM, siden andre stormaskinleverandører har gjort det nærmest siden tidenes morgen: Kjørt mange virtuelle maskiner – med like eller ulike operativsystemer og miljø forøvrig – på én og samme maskin. De virtuelle maskinene kan betjene én eller flere brukere, mens tilgjengelige hardware-ressurser bestemmer hvor mange virtuelle maskiner som kan håndteres samtidig og hvor mye ressurser de kan få tilgang til. Ineffektivt? Det kommer an på en rekke faktorer, og sett fra et driftssynspunkt er det innlysende at løsningen har mange fordeler.

Årsakene til at konseptet nå kommer til heder og verdighet igjen, er interessante. For det første er driftsaspektet like besnærende i dag som for 15 år siden. Hundrevis av selvstendige x86-tjenere i hyller bortover maskinrommet er vel og bra, og ga nødvendig robusthet og feilisolasjon i en periode hvor systemene hadde begrenset pålitelighet. Dette har forandret seg, samtidig med at kapasiteten til hvert enkelt system har vokst vesentlig. Kostnadene knyttet til hardware-ressurser – prosessorkraft, hukommelse, masselager, båndbredde – er små i forhold til programvare og driftskostnadene, hvilket snur opp-ned på regnestykker som har vært allment akseptert siden 90-tallet oppratt.

Å kjøpe et vilkårlig antall forekomster av Windows 2000 eller en annen utgave av Windows – eventuelt et helt annet operativsystem, er kort og godt ikke lenger en akademisk øvelse for å se om det er mulig, men en mulighet som stadig oftere synes attraktiv både økonomisk, teknisk og driftsmessig. VMware gjør øvelsen mulig i kommersiell forstand: Produktet emulerer en komplett Intel-maskin, med Linux eller Windows 2000 som vert. Leverandøren, som bærer samme navn som produktet, har relativt nylig gjort nettopp denne muligheten til et poeng i markedsføringen – samtidig med at en såkalt Enterprise-variant av produktet nå er tilrettelagt for nettopp slik anvendelse. Brukermiljøer rapporterer om gode resultater med opp til 20 virtuelle maskiner på samme 4-prosessor system, hvilket forteller om gode skaleringssegenskaper. Ifølge VMware er 64 virtuelle maskiner på én fysisk maskin fullt mulig, sammen med støtte for opp til 64 GB fysisk hukommelse. Fristende konsolidering? Mange mener nettopp det – også IBM, som lisensierer produktet fra VMware. Effektiviteten i systemet er påviselig overraskende god, mens det er et kjent faktum at mange Intel-tjenere er underutnyttet og kjører med belastninger i størrelsesorden 10-20%. Her er det åpenbart gevinster å hente.

Vi har tidligere diskutert VMware i Mellvik-Rapporten nr. 62, 68 og 90 – www.vmware.com

markedsføring: SFU 3.0 er full av slik programvare og kunne ikke ha eksistert uten. Mens vi ønsker en slik pragmatisme velkommen, er det vanskelig å ikke reflektere over avstanden mellom liv og lære.

Ytelse

I presentasjoner av SFU 3.0 legger Microsoft stor vekt på at ytelsen nå er sammenlignbar eller bedre enn RedHat Linux 7.2. Denne poengteringen viser blant annet hvordan selskapet evaluerer konkurransen i markedet – hvor truslene er å finne. Målsettingen er åpenbar: Å stoppe deserteringsstrømmen fra Windows til Linux på tjenersiden – der både kostnadsmessige og ytelsesmessige forhold har stilt Windows i et dårlig lys de siste 5 årene, spesielt som fil-, epost-, Web- og utskriftstjener.

Derfor har også ytelse hatt en sentral plass i forbindelse med videreutviklingen av SFU i de to årene som har gått siden forrige utgave – og resultatene er godt synlige. Subjektivt er det liten eller ingen forskjell i opplevd ytelse på tilsvarende operasjoner om vi kjører

SFU 3.0 eller RH Linux lokalt. De forskjellene som fra tid til annen viser seg, har sannsynligvis mer med vindussystemet enn med operativsystemet å gjøre. At disse to i Windows' tilfelle henger tett sammen og derfor ikke kan måles, er en virkelighet vi må avfinne oss med.

I tråd med Microsofts målsetting er imidlertid tjenerytelse langt mer interessant: Hvordan fungerer W2K med SFU 3.0 som tjener i et Unix/Linux-miljø? "Bedre enn Linux" sier Microsoft, og slår i bordet med uavhengige tester som viser at som NFS-tjener er SFU bedre enn RedHat Linux 7.2 på praktisk talt alle punkter.⁶ Forskjellen er ikke stor, men stor nok til at påstanden kan forsvares. Videre er kun NFS-ytelse testet, hvilket betyr at dette etter Microsofts oppfatning er det viktigste området – eller kanskje det eneste området som er konsistent bedre enn konkurrenten.

⁶ Testene er utført av Etesting Labs i USA og er tilgjengelig på adressen http://www.etestinglabs.com/main/reports/ms_nfs_performance.pdf.

WHERE'S THE BEEF?

Mens alt dette høres vel og bra ut, dukker det opp høyst relevante spørsmål underveis: Hvor og eventuelt i hvilke sammenhenger er det meningsfylt å benytte et kunstig miljø som dette? Kan Microsoft og Windows levere et bedre og mer kosteffektivt Unix-miljø under Windows enn Unix-leverandørene og Linux? Eller står vi overfor nok et forsøk på å utvide et universalverktøy som kan brukes til det meste, men som ikke blir bra for noe? Hvor stort er markedet og når er det fornuftig å betale Microsoft-priser for noe som er praktisk talt fritt tilgjengelig?

Utfordringene knyttet til emulering og simulering er de samme uansett hvilke plattformer som betraktes og hva som emuleres. Unix og Linux blir i beste fall en middelmådig plattform for Windows-applikasjoner, og forholdet blir det samme motsatt vei. Når vi ser på standard nettverks-tjenester blir imidlertid situasjonen en annen: De er synlige for sine brukere (klienter) utelukkende via nettverket, og kan implementeres på en måte som er optimal for plattformen – uten tanke på emulering. Derfor har det vært mulig for Linux og Unix med SAMBA (se Mellvik-Rapporten nr. 60 og IT-revyen i Mellvik-Rapporten nr. 85) å emulere NT- og W2k-tjenere i et nettverk, uten at klientene har sett noe annet enn positive (ytelsesmessige) forandringer. Mulighetene for å gjøre tilsvarende den andre veien er ikke bare til stede, men realiserte med SFU 3.0, slik ytelsesmålingene av NFS ovenfor bekrefter.

Kostnader og lisenser

I forhold til Linux, som Microsoft har utpekt til hovedmotstander, blir imidlertid Windows i alle sammenhenger stående igjen med et vesentlig handicap: Lisenskostnader, som ikke bare er høye ved anskaffelser, men som erfaringsmessig stiger over tid og dessuten er uforutsigbare, selv for store kunder. Først er selve operativsystem-lisensen betydelig, dernest kommer i de fleste sammenhenger en per bruker kostnad, og til slutt SFU – til USD 99 eller ca. NOK 1.000 per tjener. Her må det åpenbart grundige kalkulasjoner til for å bringe balanse i regnestykket. TCO – *Total Cost of Ownership* – er metoden, og det er ganske innlysende at miljøer der Windows-tjenere er dominante kan tjene på å forenkle – homogenisere – maskinparken sett fra et driftsmessig ståsted. Imidlertid er mange av faktorene som inngår i ligningen vanskelige å kvantifisere, og kalkulasjonenes pålitelighet påvirkes naturligvis av dette.

Videre er det rimelig å anta at en konsolidering av flere tjenester på samme tjener vil virke positivt for både TCO og andre økonomiske forhold, og at SFU kan benyttes i en slik sammenheng. I den tekniske dokumentasjonen advarer imidlertid Microsofts eksperter mot slike kombinasjoner: Det er best å ha dedikerte systemer til SFU-tjenester, understrekes det. Dermed dukker spørsmålet opp på nytt: Er vi ikke bedre tjent med en 'ekte' tjener for de tjenester og applikasjoner det er snakk om?

Vi kan ta for gitt at Microsoft, som vi var inne på innledningsvis, ønsker å finne veier inn i nye markeder der Unix/Linux er fullstendig

dominant. SFU 3.0 er uten tvil et viktig redskap i så henseende, men vil i første omgang gjøre livet tøffere for konkurrentene på emulator-siden – med MKS i spissen. Forholdet er nøyaktig det samme som vi ser på Unix/Linux-siden: Linux' suksess merkes i første omgang vel så mye av Unix-leverandørene som av Microsoft.

Konklusjon

På Networld+Interop i Las Vegas i mai slo Microsoft virkelig på stortrommen med lanseringen av SFU 3.0: Seminarer, *workshops*, presentasjoner, CDer, dokumentasjon og mer i tre dager til ende. Inklusive personalkostnader kom regningen ganske sikkert på et 6-sifret beløp – i USD. Småpenger for en gigant som tjener milliarder ukentlig, men likevel en oppsiktsvekkende satsing på et nisjeprodukt som i seg selv neppe noen gang vil bidra direkte til Microsofts nettoresultat.

SFU kunne lett ha gått over i historien som et komma – i likhet med sine forgjengere, men unngår denne skjebnen fordi Microsoft vil mer med produktet enn å tilfredsstille enkelte smale markedssegmenter. Produktet har kvalitetsmessige sider som uten tvil fortjener positiv oppmerksomhet, men dette er langt fra nok til å skape en suksess. Problemet med SFU er først og fremst at markedet er minimalt. Om produktet er aldri så godt, forblir det meningsløst å investere tusenvis av kroner i programvarelisenser for et produkt som emulerer Unix, når et like godt eller bedre produkt er tilgjengelig praktisk talt gratis.

Microsoft vet naturligvis dette bedre enn de fleste. Selskapet ble ikke verdens mest lønnsomme ved å kaste penger ut av vinduet. Vi må lete andre steder etter selskapets motivasjon for å investere så kraftig i et produkt som neppe noen gang blir i stand til å stå på egne ben. At denne motivasjonen sannsynligvis er av strategisk art, kreves det verken krystallkule eller overnaturlige evner for å komme frem til. All utvikling tar tid, trender begynner i det små og vokser sakte før de eventuelt eksploderer.

Strategi og posisjonering

Microsoft har lang erfaring i å oppdage og forberede seg på – eller i mange tilfeller ta styringen av – trender under utvikling. Vi konstaterte innledningsvis at selskapet gjentagne ganger i løpet av de to siste årene har utpekt Linux til sin viktigste motstander og største trussel. Aktiviteter for å motarbeide denne trusselen er i gang på mange felter, og SFU er en av dem. Det handler om politikk, posisjonering, taktikk og planlegging.

I det ene landet etter det andre – primært i Europa, Sør-Amerika og Asia – bestemmer offentlige etater og regjeringer seg i disse dager for å gå andre veier enn de som er staket opp av Bill Gates, Steve Ballmer og deres medarbeidere. *Open Source* – og i noen tilfeller Linux – blir introdusert, i noen tilfeller som alternativ, i andre tilfeller som et krav. Ønsket er å skape mer konkurranse, valgmuligheter og lavere kostnader, og Microsoft er i mange tilfeller den stygge ulven.

Inntil nylig hadde selskapet lite å stille opp med i slike tilfeller. Kravet om *Open Source* utelukket Microsoft og Windows. Med SFU ser bildet plutselig annerledes ut: Ikke bare kan Windows påviselig opptre som vert for store deler av verdens *Open Source* programvare, men sågar levere noen av tjenestene like effektivt som Linux. Windows er ikke lenger utelukket fra å delta i anbudskonkurranser der kravet er støtte for nye utgaver av POSIX/X-Open-standardene eller *Open Source*. Der det er hensiktsmessig kan Microsoft sågar slå seg på brystet med at gehalten av nettopp *Open Source* i SFU 3.0 allerede er betydelig.

Ingen kan kritisere selskapet for dette. Tvert imot tar vi av oss hatten for både kreativitet og fremsynthet, samtidig som vi konstaterer at kvaliteten på produktet som leveres er god. Derfra til å mene at det er en god idé å kjøpe personbil og tilhenger når vi egentlig trenger en lastebil, er imidlertid avstanden lang.

Trussel for Citrix?

En langt mindre åpenbar, men ikke desto mindre rimelig bakenforliggende strategi for SFU, er å ta tilbake deler av markedet for Windows-baserte flerbruker-systemer. Microsofts 'shanghaiing' av Citrix' opprinnelige flerbruker-teknologi, som i dag er innbakt i TSE-utgaven av W2k, har vært viktig for Microsoft, men har ikke forhindre Citrix i å blomstre med ekstra funksjonalitet, tjenester og verktøy som Microsoft ikke kan levere. Situasjonen er den at de fleste store TSE-installasjoner verden rundt velger Citrix-produkter på toppen av Windows for å få en flerbrukerplattform med akseptabel funksjonalitet, effektivitet og driftsverktøy.

Citrix fornyet nylig sin lisensavtale med Microsoft, som gir selskapet tilgang til kildekode for tjenerutgaver av Windows i ytterligere 4 år. Microsoft benyttet ikke sin mulighet til å kjøpe eller lisensiere Citrix' teknologi slik de gjorde ved forrige korsvei. Dette kan bety at Gates & Co. ikke er interessert i dette markedet, eller at de har andre planer. Sannsynligheten taler for det siste, at Microsoft vil replisere store deler av Citrix' NFuse-teknologi med utgangspunkt i SFU, som bringer Windows et stort steg i retning av et ekte flerbruker-system. Det tekniske grunnlaget er godt og utfordringene definitivt overkommelige. Dessuten vil en slik 'vri' uten videre bidra til å gjøre investeringene i SFU langt mer rimelige sett fra et kommersielt synspunkt.

BUSINESS AS USUAL

I første omgang forblir imidlertid SFU eksakt hva den gir seg ut for å være: Et hjelpemiddel for Microsoft til på den ene siden å tilfredsstille kompatibilitetskrav, og på den andre siden å møte konkrete behov hos kunder som ønsker homogenisering og standardisering på Windows-plattformen.

Om vi ser bort fra de kostnadmessige innvendingene, fyller SFU denne ambisjonen på en fortreffelig måte, og blir en effektiv brobygger mellom disjunkte miljøer – for programmer og ikke minst for drift: Den gir mulighet til å drifte Windows-systemer som om de var Unix-systemer.

mer og motsatt, hvilket representerer oppfyllelsen av en drøm for tallrike driftsmiljøer over store deler av verden – en inkarnasjon av Ole Brumm-filosofien: Ja takk, begge deler! ■

Kopiering forbudt