Laborationer i kursmomentet Datoranvändning E1

Laboration nr 8: Nätverk och prestanda

98-11-08 Thomas Hallgren

1 Introduktion

På de flesta ställen där det finns mer än en dator är datorerna hopkopplade i ett lokalt nätverk. UNIX erbjuder ett sätt att använda nätverk som inte brukar förekomma i PC/Macintosh-nätverk, nämligen möjligheten att köra program på andra datorer än den man själv sitter vid (på engelska *remote execution*). I denna laboration utnyttjar vi denna möjlighet för att enkelt köra ett benchmark (ett program som testar hur snabbt en dator löser en viss uppgift) på flera olika datorer. Övningarna belyser också att fönstersystemet X Windows (dvs det fönstersystem som är standard i UNIX) är *nätverkstransparent*, dvs vilken dator programmet körs på och vilken bildskärm programmets fönster visas på kan väljas oberoende av varandra, samt att det inte alltid lönar sig att köra ett program på en annan dator, även om den är snabbare.

Kommandon och program som dyker upp i denna laboration är: `rlogin`, `slogin`, `logout`, `ps`, `ping`, `traceroute`, `xmfract` och `xaos`.

2 Uppgifter

Som tidigare markerar symbolen ☐ en konkret arbetsuppgift.

Skriv ner svaren på frågorna och visa dem för en handledare för att bli godkänd.

2.1 Uppgifter från förra veckan

☐ Laboration nr 7 tog lite mer an två timmar att göra för de flesta, så om du inte hann göra klart alla uppgifterna, gör det nu.

2.2 Att köra program på andra datorer

Några vanliga tjänster som lokala nätverk brukar erbjuda är:

- skrivardelning (man kan ha en gemensam skrivare, istället för en per dator).
- fildelning (filer som finns på hårddisken i en dator är tillgängliga på andra datorer).

I UNIX-system brukar detta vara gjort så att man inte märker att nätverket finns. Alla datorerna verkar likadana, dvs alla filer, program och skrivare är tillgängliga på alla datorerna, utan att man behöver veta vilken dator de egentligen finns på.

Anm. Att det är så är ju väldigt bekvämt. Det är antagligen en kvarleva från tiden då persondatorer, arbetsstationer och nätverk inte fanns och man istället bara hade en enda, stor och klumpig dator med flera textterminaler inkopplade via långa sladdar. Då fanns alla filer på samma dator och alla terminalerna fungerade likadant.

I typiska nätverk med UNIX-datorer spelar det alltså ingen större roll vilken dator man sitter och arbetar vid, när det gäller vad man har tillgång till. Datorerna kan dock skilja sig på ett antal punkter:

- De kan ha olika bra bildskärmar, tangentbord och mus.
- De kan ha olika prestanda, t ex vara olika snabba och ha olika mycket primärminne.

På många arbetsplatser står det nuförtiden en dator på varje skrivbord. Ett sätt att hålla nere kostnaderna, samtidigt som man ger alla användare tillgång till en kraftfull dator för krävande beräkningar, är att ha enkla, billiga datorer på skrivborden och en gemensam snabb, men dyr dator åtkomlig via *remote execution*. Detta fungerar ofta bra, eftersom det typiska är att en användare bara utnyttjar datorns kapacitet en liten bråkdel av tiden.

Anm. Detta sätt att bygga upp datormiljön påminner lite om det gamla sättet, dvs en enda dator som används via enkla textterminaler.

I UNIX-system brukar det vara väldigt enkelt köra program på vilken dator som helst i det lokala nätverket. Man använder sig av något av kommandona `rsh`, `ssh`, `telnet`, `rlogin` eller `slogin`. De två första är lämpliga för att köra ett enstaka kommando, de tre följande är lämpliga när man ska köra flera kommandon, eller interaktiva program. `ssh` (secure shell) och `slogin` (secure login) är efterföljare till `rsh` (remote shell) respektive `rlogin` (remote login), som har UNIX-bakgrund. `telnet` är ett mera systemoberoende kommando.

Uppgift 1. Vi testar några datorer med Dhrystone-testet

Hur snabba är de olika datorerna på E-sektionen? Vi provar det genom att köra Dhrystone-testet, som är ett gammalt enkelt benchmark (dvs program för att jämföra prestanda), som mäter hur snabb en dator är på heltalsberäkningar och enkel texthantering.

Dhrystone-programmet finns i filen `~hallgren/Intro/dhry`.

Programmet frågar hur många gånger man vill att testet ska köras. På våra datorer är 1000000-2000000 lagom. (När programmet skrevs räckte det kanske med 10000 gånger.) Tar man för få klagar programmet på att mättiden blir kort för att resultatet ska vara pålitligt.

☐ Provkör `dhry` på den dator du sitter vid. Vilken dator är det? ___________________

När testet är klart skrivs en del kontrolltexter ut. Sist kommer resultatet, antalet Dhrystone per sekund. (Programmet mäter själv tiden.) Vad blev resultatet för den dator du sitter vid? _____________

☐ Provkör programmet `dhry` på några fler datorer och fyll i Tabell 1. Använd kommandot `rlogin` *datornamn* för att "logga in" på de olika datorerna. Avsluta med kommandot `logout` när du är klar med varje dator.

Dator	Dhrystones per sekund
`quarl32`	
`hortensia`	
`quarl102`	
`quarl202`	

Tabell 1: Resultat av Dhrystone-testet på några av E-sektions datorer

Anm. En detalj att lägga märke till när man använder `rlogin` är att man inte behöver tala om vem man är innan man släpps in på den andra datorn. Detta beror på att `rlogin` automatiskt skickar över ditt användarnamn till den andra datorn och att datorerna i det lokala nätverket litar på varandra.

2.3 Att köra fönsterprogram på andra datorer

Vi har nu sett att program som körs i ett terminalfönster lätt kan köras på en annan dator än den man sitter vid. Hur går det med program som öppnar egna fönster? Fönstersystemet i UNIX, X Windows, är *nätverkstransparent*, dvs det spelar ingen roll om bildskärm, mus och tangentbord (som tillsammans utgör en *X-terminal*) finns på samma dator som programmet körs på, eller om det är ett nätverk mellan dem.

Valet av X-terminal styrs av omgivningsvariabeln `DISPLAY`. Man kan använda kommandot `echo $DISPLAY` för att se hur den är definierad och `setenv DISPLAY ...` för att definiera om den. (Vi har tidigare stött på omgivningsvariabeln PRINTER, som på samma sätt styr valet av skrivare). Det vanliga är dock att man vill ha fönsterna på den dator man sitter vid, så man behöver inte bry sig så mycket om detta.

När man använder `rlogin` eller `telnet` för att logga in på en annan dator blir inte `DISPLAY`-variabeln definierad automatiskt. Det kan också av säkerhetskäl hända att det inte räcker att ställa in den rätt heller. Kommandot `slogin` löser dessa båda problem på ett smidigt sätt, så vill man köra fönsterprogram på en annan dator är det lämpligt att logga in med `slogin`.

`ssh` och `slogin` har ett säkerhetssystem som baseras på kryptering med s k öppna nycklar. Sådana skapas första gången man använder något av kommandona och ett meddelande som upplyser om detta dyker upp. Det dyker också upp en fråga första gången man kopplar sig till en viss dator.

Uppgift 2. Kör fönsterprogram på olika datorer

Programmet `xmfract`, som visar fraktaler, finns i `~hallgren/Intro/`. Vi provar att köra det på den långsammaste och den snabbaste datorn enligt Tabell 1.

☐ • Ta fram ett terminalfönster och logga in med `slogin` på den långsammaste datorn. Starta programmet `xmfract` i bakgrunden (genom att skriva ett `&`-tecken sist på kommandoraden). Klicka bort eventuella meddelanderutor som visas.

- Tag fram ett nytt terminalfönster och logga in på den snabbaste datorn. Starta `xmfract` på samma sätt som ovan.

Välj menykommandot **Image/Create Image** i båda `xmfract`-fönstren. Programmen ritar nu upp en bild av Mandelbrot-mängden. Förhoppningsvis märker man nu en viss skillnad i hastighet på de båda datorerna.

Kommandot `ps` visar information om de program man har startat. Kör det båda terminalfönstren och leta upp `xmfract` i listan. Kolumnen `TIME` visar hur mycket processortid programmet har använt.

- Hur mycket processortid behövdes för att generera bilden på den långsamma datorn? ________________
- Hur mycket tid behövdes på den snabba datorn? ___________
- Stämmer skillnaden i tid med hur snabbt bilden ritades enligt din uppfattning? ___

Det är ju inte bara hur snabb processorn är som avgör hur lång tid det tar att utföra en viss uppgift. Om du svarat ja på ovanstående fråga är det förstås processorn som har störst inverkan i det här experimentet. Om du svarat nej är det andra faktorer. En sådan faktor kan vara nätverkets prestanda.

2.4 Att undersöka nätverkets prestanda

När lönar det sig att köra ett program på en annan dator som är snabbare än den man sitter vid? Program kan delas in i *CPU-intensiva* och *IO-intensiva*, dvs program vars körning domineras av beräkningar respektive in- och utmatning. Det finns förstås också mellanting. Ett program som är CPU-intensivt, som t ex Dhrystone-testet, vinner förstås på att köras på en snabbare dator. Ett program som är IO-intensivt, som t ex ritar mycket i sitt fönster, visar stora bilder eller film, kanske blir långsammare om man kör det på en annan dator. Nätverkets prestanda har då större betydelse en datorns.

Två parametrar bestämmer ett nätverks prestanda. Den första är fördröjningen: hur lång tid tar det för ett meddelande som skickas genom nätverket att komma fram? Den andra är kapaciteten: hur mycket data kan överföras på en viss tid?

Uppgift 3. Mät fördröjningar med ping

Kommandot `ping` *`datornamn`* skickar ett meddelande till en dator, denna skickar genast (förhoppningsvis) ett svar. Detta gör att man kan mäta fördröjningarna i nätverket.

☐ Undersök hur stor fördröjningen är vid kommunikation mellan den dator du sitter vid och några andra datorer och fyll i kolumnen Ping-tid i Tabell 2. Kommandot `ping` skickar ett meddelande i sekunden till man avbryter det med CTRL-C. Man kan också begränsa antalet meddelanden t ex till 2 genom att skriva `ping -c 2` *`datornamn`*.

Kategori	Datornamn	Ping-tid	Hopp
En annan dator på E	`hortensia`		
En dator någon annanstans på Chalmers	`dogbert.cs.chalmers.se`		
En dator i norra Sverige	`www.luth.se`		
En dator i USA	`www.yahoo.com`		
En dator i Japan	`www.yahoo.co.jp`		

Tabell 2: Mätning av fördröjningar och avstånd i nätverket

Uppgift 4. Undersök nätverkets kapacitet

Istället för att försöka mäta hur stor överföringskapacitet nätverket på E har gör vi ett praktiskt prov.

Program som visar rörliga bilder skickar stora mängder data till sitt fönster. Ett räkneexempel: hur många megabyte per sekund behöver överföras då 20 bilder per sekund visas i ett fönster som är 480×360 pixels stort (med 1 byte per pixel)? __________

☐ Programmet `xaos`, som också det finns i `~hallgren/Intro/`, ritar mandelbrotmängden så snabbt att man kan zooma i den i realtid. Starta `xaos` på den dator du sitter vid *och* på en annan, minst lika snabb dator. Prova att zooma i de båda fönstren genom att hålla vänster musknapp nertryckt (man zoomar ut med höger musknapp). Du märker förhoppningsvis en viss skillnad mellan fönstren. Beskriv skillnaden: ______________________________

Lönar det sig att köra detta program på en annan, snabbare dator? ______________

Uppgift 5. Vägar genom Internet

☐ Internet kopplar ihop lokala nätverk på olika platser i världen. Meddelanden som skickas via Internet passerar normalt många kopplingspunkter (routers) på vägen. Kommandot `traceroute` *`datornamn`* visar vilken väg genom nätet meddelanden till en viss dator skickas. Kör det och undersök hur många hopp meddelanden gör (hur många ställen det landar på, inklusive målet) när de skickas till datorerna i Tabell 2.

3 Avslutande kommentarer

Remote execution tillåter att man använder en kraftfull dator utan att man behöver vara i närheten av den. Man t ex sitta vid en dator på E-sektionen och köra simuleringar på en supersnabb parallelldator på andra sidan jordklotet.

Innan man startar ett program på en annan dator kan det förstås vara lämpligt att kontrollera om någon annan redan använder datorn. Några användbara kommandon är:

- `v` - visar vilka datorer som är lediga (finns endast på E-sektionen).

- `who` - visar vilka som är inloggade på den dator kommandot körs på.
- `w` - visar vilka program användare på datorn kör. (`ps` kan fås att visa mer.)
- `uptime` - visar medelvärden över hur hårt belastad datorn har varit den senaste tiden. 0.00 = datorn gör inget, 0.50 datorn har arbetar med något 50% av tiden, 1.00 = datorn är upptagen med något hela tiden, 2.00 = datorn har i genomsnitt två saker olika saker att göra och växlar mellan dem, o s v.

Kom ihåg att `man`-kommandot ger detaljerade beskrivningar av kommandon.

3.1 Vad händer när man kör två program samtidigt på en dator?

Hur bra det går att köra två program samtidigt på en dator beror på vilka resurser i datorn de använder. Man kan i princip köra ett CPU-intensivt program samtidigt med ett IO-intensivt program samtidigt utan att något av dem störs av det andra.

Kör man flera CPU-intensiva program samtidigt fördelas processorns tid rättvist mellan alla programmen. Processorn kan tämligen snabbt växla mellan olika beräkningar, så det går ungefär lika fort att köra två program samtidigt som att köra det ena efter det andra.

Hårddiskar är däremot inte så snabba på att växla mellan att läsa/skriva på olika ställen på disken, så om man kör två program som båda läser/skriver stora filer kan det gå betydligt långsammare om man kör dem samtidigt istället för ett i taget.

Det blir också problem om program som körs samtidigt tillsammans behöver mer minne än det finns i datorn. Tack vare det virtuella minnessystemet kan programmen fortsätta köra, men data kommer att kopieras fram och tillbaka mellan primärminne och den s k swaparean på hårddisken, vilket tar mycket tid, och datorn kan bli väldigt långsam.

3.2 Information på webben

På CPU Info center, `http://infopad.eecs.berkeley.edu/CIC/`, hittar man massor med information om olika processorer, bland annat en lista med benchmarkresultat.

SPEC, The Standard Performance Evaluation Corporation, finns på `www.spec.org`. De presenterar bland annat resultaten av sina benchmark (SPECint95, SPECfp95, m fl).

På `http://www.top500.org/top500.list.html` finns en lista på världens 500 snabbaste datorer. Två av dem finns på Chalmers.

På `http://www.internettrafficreport.com/` visas med olika diagram hur hårt Internet är belastat.

Information om `ssh` (och `slogin`) hittar man om man går till `www.yahoo.com` och fyller i `ssh` i sökrutan. Det finns gratisversioner för Unix och Windows.

En lathund över kommandon som använts i laborationerna kommer i veckan att dyka upp på kursens hemsida.