



door Guido Socher (homepage)

Applicaties op afstand draaien met X11



Over de auteur:

Guido houdt van Linux en niet alleen omdat het interessant is te leren hoe een besturingssysteem werkt, maar ook vanwege de mensen die betrokken zijn bij het ontwerp ervan.

Kort:

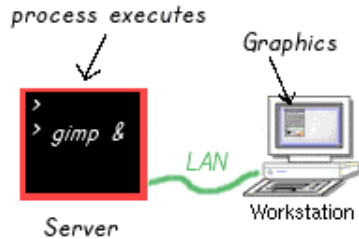
Veel mensen die Linux voor het eerst gebruiken denken dat het grafische bureaublad onder Linux weer zo'n "Windows" systeem is waarin je applicaties kan starten die in gescheiden vensters verschijnen. Sommigen merken op dat je ook verschillende bureaubladen kan hebben, maar verder dan dat gaat het niet. Het Linux X Window System (X11) kan nog veel meer! Het is een window systeem voor op het netwerk. We zullen zien welke nieuwe en krachtige oplossingen dit biedt.

*Vertaald naar het
Nederlands door:
Hendrik-Jan Heins
<hjh/at/passys.nl>*

Het display concept

Iedere grafische X Window applicatie leest bij het opstarten de omgevingsvariabele DISPLAY om uit te vinden naar welke computer monitor het z'n beelden moet sturen. Dit tezamen met de netwerkmogelijkheden van het X Window System maakt het mogelijk om grafische applicaties op afstand te draaien. Dit houdt in dat je de CPU van de ene machine gebruikt terwijl je de applicatie aanstuurt vanaf een ander. De volledige GUI (graphical user interface ofwel grafische gebruiksinterface) verschijnt op de machine waar vanaf je hem aanstuurt. Je merk niet dat je eigenlijk op 2 computers tegelijk werkt.

De snelheid van het netwerk is hierbij natuurlijk van belang, maar een gewoon 10Mbit/s LAN netwerk is meer dan snel genoeg.



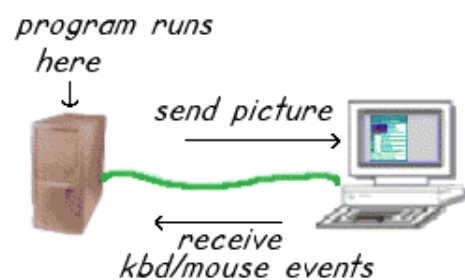
Waarom zou je dit willen doen?

Er zijn veel mogelijke toepassingen voor deze "network graphics". Bedrijven gebruiken ze om op afstand apparatuur aan te sturen die misschien wel duizenden kilometers verderop staat en jij kan de toepassingen van al je computers gebruiken alsof je er zelf achter zit.

Je zou 2 computers kunnen hebben, een snelle 1Ghz machine en een oude Pentium 133MHz. Je kan gebruik maken van de snelheid van je nieuwe machine zonder dat je erachter zit. Misschien zit je zus nu achter de snelle computer en is ze ingelogd en aan het werk. Het maakt niet uit, je kan er nog steeds gebruik van maken.

Hoe werkt het?

Alle X Window applicaties, of ze nu gimp, xterm, konqueror of netscape heten ... zijn in feite netwerk clients die verbonden worden met een server, de X-server. Het is de taak van de X-server om te communiceren met de grafische hardware, de plaatjes op je scherm te tekenen en de muis- en toetsenbord-invoer te lezen. De clients (de cliënten, programma's zoals gimp, netscape...) sturen de server instructies over hoe de frames en de knoppen getekend moeten worden. In ruil krijgen zij van de server de gegevens van de muis en het toetsenbord. Je hebt natuurlijk één of andere authenticatie nodig anders zou iedereen op ieders scherm kunnen komen. Er bestaan 2 programma's die de toegang beheren:



- **xhost**: Dit programma laat je toe gebruikers toestemming te geven om op een bepaalde machine grafische gegevens op het scherm te zetten. Bijvoorbeeld: Je zit achter een machine die philosophus heet. Om toegang te verlenen voor ieder programma vanaf de machine movietux op jou scherm (op philosophus) zou je het volgende commando moeten typen:

```
xhost +movietux
```

Dit moet op een commandoregel op philosophus worden getypt

- **xauth**: Dit is een op cookies gebaseerde authenticatie en hij is veel geavanceerder. Hiermee kan je echt individuele mensen toegang geven. Het is ook veel beter beveiligd dan xhost. De authenticatie verloopt via een cookie dat bewaard wordt in het bestand .Xauthority in de "home" directory van een gebruiker. Als de op een andere locatie staande machine een geldig cookie bevat in dit bestand, dan wordt er toegang verleend. Om een cookie te kopiëren vanaf de machine waar je achter zit (philosophus) naar de andere die het programma draait (movietux) kan je één van de volgende methodes gebruiken:

```
xauth extract - philosophus:0.0 | ssh movietux /usr/X11R6/bin/xauth merge
```

of
scp ~/.Xauthority movietux:

De procedure die start bij het opstarten van een programma (client) ziet er als volgt uit:

1. De client controleert de DISPLAY omgevingsvariabele om de server te vinden en probeert daarmee een verbinding te maken. Lukt dit niet (of is er geen server op afstand opgegeven), dan maakt de client verbinding met de lokale server.
2. De server controleert of de client toestemming heeft om "beelden" naar hem toe te zenden. Als de client toestemming heeft zal de server het beeld weergeven op het scherm.

De DISPLAY omgevingsvariabele heeft de volgende syntax:

```
bash:export DISPLAY=hostnaam:displaynummer.schermnummer  
tcsh:setenv DISPLAY hostnaam:displaynummer.schermnummer
```

Ik zal niet ingaan op het displaynummer en schermnummer in dit artikel. Dit is normaal gesproken gewoon 0:0. Een voorbeeld voor bash zou er als volgt uit zien:

```
export DISPLAY=philosophus:0.0
```

Een applicatie op afstand starten

Na alle theorie nu een praktisch voorbeeld. Probeer het gewoon eens. We zitten weer achter de computer philosophus en we willen iets op afstand opstarten vanaf movietux.

1. Vertel aan je X-server dat clients vanop movietux beelden op je scherm mogen zetten:
xhost +movietux
2. login op de op afstand staande gastheer movietux:
slogin -l username movietux
3. Nu ben je ingelogd op movietux en ieder commando dat je ingeeft wordt uitgevoerd op movietux
4. export DISPLAY=philosophus:0.0
5. Start een programma. Bijvoorbeeld:
netscape &

Als movietux een snelle machine is dan zal je zien dat html pagina's sneller worden gerenderd (opgebouwd) dan op de machine waar je achter zit. Tegelijkertijd zie je niet direct dat netscape niet lokaal gestart is, aangezien je het nog op precies dezelfde manier met je muis en toetsenbord aanstuurt. Cool, niet?

Het scherm met je meenemen

Ook al is het eenvoudig om een applicatie op afstand te draaien, er is nog steeds een beetje extra typwerk nodig om het DISPLAY in te stellen. Het is ook mogelijk om dit automatisch te laten verlopen:

- Wanneer je gebruik maakt van ssh om in te loggen op de gastheer op afstand, dan wordt het DISPLAY automatisch goed gezet. Er zijn nog andere programma's om mee in te loggen vanop afstand die DISPLAY ondersteunen, maar ssh wordt zeer veel gebruikt.
- Als je gebruik maakt van slogin of andere commando's dan kan je het onderstaande script installeren op de gastheer op afstand.

Voor tcsh:

```
# neem je display mee bij een login op afstand:
# Zet het in je ~/.login bestand
set whoami='who -ml'
set remhost='expr "$whoami" : '.*(\.*)''
if ( "$remhost" != "" ) then
setenv DISPLAY "$remhost":0.0
endif
```

Het script werkt door de naam van de gastheer op afstand op te vragen door middel van het commando "who -ml". Dit commando zal iets geven als:

```
>who -ml
movietux!guido pts/3 Oct 26 21:55 (philosophus.tux.org)
```

als je bash gebruikt heb je het volgende script nodig:

```
# neem je display met je mee naar een login op afstand:
# Plaats dit in je ~/.bash_profile bestand
whoami='who -ml'
remhost='expr "$whoami" : '.*(\.*)''
if [ -n "$remhost" ]; then
DISPLAY="$remhost":0.0
export DISPLAY
fi
```

OpenGL

Terwijl de netwerk mogelijkheden van het X Windows Systeem zeer uitgebreid zijn, is de grafische prestatie wat minder doordat je de gegevens over een netwerkprotocol zendt. Maar normaal gesproken zal je niet veel verschil zien.

Grafisch zware en snelle applicaties zoals vele spellen zijn meestal gebaseerd op OpenGL (Open Graphics Library) en GLX (OpenGL Extentie voor het X Window System). Deze bibliotheken hebben een hardware-onafhankelijke programmeer-interface die directe toegang tot 3D hardware versnelling op de grafische kaart mogelijk maakt. Dit houdt in dat de applicatie de omschrijving

van een object in de vorm van punten, lijnen en polygonen naar de grafische kaart stuurt en dat alle rendering vervolgens wordt afgehandeld door de grafische kaart. Dit maakt zeer snelle beelden mogelijk.

Op dit moment is het met de meeste drivers voor grafische kaarten (= X servers) onder Linux niet mogelijk om op afstand gebruik te maken van GLX/OpenGL hardware versnelling. Ze ondersteunen alleen hardware versnelling voor lokale applicaties. Het resultaat hiervan is dat op afstand gedraaide OpenGL applicaties nauwelijks starten en zeer langzaam zijn. Een uitzondering hierop vormen de NVidia drivers (waarvoor de broncode niet is vrijgegeven). Zij hebben een direct rendering interface die indirecte rendering voor applicaties op afstand mogelijk maakt.

Conclusie

Gebruik maken van de rekenkracht van je netwerk is zeer eenvoudig met X11. Je kan op afstand werken met verschillende applicaties net zoals je werkt met lokale applicaties. Het enige verschil dat je tegen zal komen is dat je de bestanden en de "home" directory op de op afstand staande gasteer terugvindt. Als je echter NFS en NIS hebt geïnstalleerd kan je zelfs dit kleine verschil verbergen en kan je zonder nadenken de volledige rekencapaciteit van de snelste machines in je netwerk gebruiken.

Links

Het X11 systeem: x.org.

Xfree86.org, het X11 systeem dat onder Linux gebruikt wordt

Site onderhouden door het LinuxFocus editors team	Vertaling info:
© Guido Socher	en --> -- : Guido Socher (homepage)
"some rights reserved" see linuxfocus.org/license/	en --> nl: Hendrik-Jan Heins < hjh/at/passys.nl >
http://www.LinuxFocus.org	