Mozilla Optimalisatie Mini-HOWTO

Salvador J. Peralta

speralta at willamette dot edu

Vertaald door: Ellen Bokhorst

bokkie at nl dot linux dot org

In dit document wordt besproken hoe de configuratie en aanpassingen op broncodeniveau kunnen worden geoptimaliseerd om Mozilla geschikter te maken als een primaire browser voor Linux en X Window. Het is niet bedoeld als een leidraad voor het programmeren van Mozilla, noch is het een handleiding in XUL.

De technieken beschreven in dit document zijn geïmplementeerd en getest in een kiosk-achtige publieke computeromgeving met client werkstations waarop Redhat Linux met XFree86 draait.

Inhoudsopgave

1. Copyright

2. Introductie

Mozilla is een Open Source, cross-platform, browser en grafische toepassingsomgeving. Op veel op Linux gebaseerde desktops groeit het uit tot een veelbetekenende component vanwege zijn stabiliteit, robuuste set met features, grote ontwikkelaarsbasis, en configuratiegemak. Deze mini HOWTO gaat over het patchen en configureren van de broncode van Mozilla om het beter te optimaliseren voor de Linux desktopomgevingen, en het is bedoeld te dienen als een algemene instructieset voor de Linux gemeenschap over hoe Mozilla te configureren en compileren.

De meest recente versie van deze Mini HOWTO kan in HTML formaat worden verkregen vanaf www.willamette.edu/~speralta/tldp/mozilla (http://www.willamette.edu/~speralta/tldp/mozilla).

3. Haal de broncode op

Er zijn voors en tegens bij het zelf compileren van Mozilla. De grootste keerzijde is dat de broncodeversie gecomprimeerd, in de meeste uitgaven een bestand is van meer dan 30 MB wat ongeveer driemaal groter is dan de binaire distributie. Een tweede nadeel is dat het compileren van Mozilla op een redelijk nieuwe machine zoals mijn 1.6 ghz Pentium 4 met 256MB geheugen, afhankelijk van eventuele andere jobs die op het systeem draaien, ergens tussen de anderhalf uur tot vier uur kan duren. Op een oudere machine, kan de compilatie van Mozilla vierentwintig of langer duren. Het grootste voordeel is dat er een groot aantal compile-time opties zijn en dat er patches kunnen worden toegepast die de performance en featureset van de browser dramatisch zal verbeteren. Deze patches zijn gewoonlijk nodig als je van plan bent Mozilla in een productie-omgeving uit te voeren.

Sla alsjeblieft de volgende paar secties over en kijk naar de Paragraaf 7,Configuratie na de installatie als je niet bent geïnteresseerd in aanpassingen van de Mozilla browser op broncodeniveau.

Op het moment dat dit document werd geschreven, was de laatste versie van de Mozilla broncode versie 1.2.1, welk via ftp kan worden opgehaald vanuit ftp.mozilla.org/pub/mozilla/releases/mozilla1.2.1 (ftp://ftp.mozilla.org/pub/mozilla/releases/mozilla1.2.1). De andere stabiele release, 1.0.1, kan via ftp worden opgehaald vanuit ftp.mozilla.org/pub/mozilla/releases/mozilla1.0.1 (ftp://ftp.mozilla.org/pub/mozilla/releases/mozilla1.0.1).

Bekijk voor een CVS-checkout alsjeblieft de documentatie op de website van Mozilla (http://mozilla.org).

4. Patch de broncode

Een niet gepatchte Mozilla is geen al te goede burger. Je kunt bijvoorbeeld als opdrachtregeloptie geen X en Y geometrieposities specificeren. Dit is op een typische eindgebruikersdesktop niet zo'n probleem, maar in een bedrijfs- of publieke computeromgeving is het een killer. Ter correctie moeten we een patch toepassen op de C++ broncode in `embedding/components/windowwatcher/src/nsWindowWatcher.cpp` . De eerste stap bestaat uit het ophalen en toepassen van de patch (http://www.willamette.edu/~speralta/mozilla.kludge.txt) geschreven door Robert Riches. Deze patch is getest op de Mozilla versies 1.0 tot en met 1.2.1.

Plak het voor het toepassen van de patch simpelweg in `embedding/components/windowwatcher/src/nsWindowWatcher.cpp`, in je Mozilla broncodestructuur. De notatie maakt gebruik van "+" symbolen om aan te duiden dat de code moet worden ingevoegd. Deze symbolen moeten worden verwijderd voordat het tekstbestand `nsWindowWatcher.cpp` wordt opgeslagen.

Stel voor het gebruiken van de patch een omgevingsvariabele, genaamd MOZILLA_SCREEN_POS in op de juiste coördinaten. Typ onder Bash bijvoorbeeld export MOZILLA_SCREEN_POS='screenx=1,screeny=1' wat de linkerbovenhoek van je browser zal instellen op de linkerbovenhoek van je scherm.

5. Configureer de broncode

Een ander probleem bij het gebruik van Mozilla in een persoonlijke of productie-omgeving is dat het code bevat die een homepage zal overschrijven die je op geregelde basis instelde, of wanneer een nieuwe release van het product wordt gedistribueerd.

Aangezien deze informatie in een configuratiebestand is opgeslagen, is het 't eenvoudigst zonder de browser in de weg te zitten het configuratiebestand te wijzigen te wijzen op standaard homepage-instellingen naar één die consistenter is met de omgeving die je gebruikt.

Mozilla bewaart de meeste compile-time configuratie op in bestanden met een .properties extensie. Je kunt deze bestanden gebruiken om tekst te specificeren in message dialogues, fonts, en andere opties. In dit geval, in de veronderstelling dat je locale en taalinstellingen en-US zijn, is het .property bestand dat je nodig hebt `region.properties` welke is te vinden in de directory `xpfe/browser/resources/locale/en-US/`.

De wijzigingen die je aan dit bestand moet aanbrengen, zijn tamelijk recht-toe-recht-aan. Als algemene regel vervang ik het waarde deel van het naam/waarde paar naar één van mijzelf bij elke instance een keuze maken waar het woord "homepage" in de configuratie wordt vermeld.

```
# navigator.properties
homePageDefault=http://jeurl.com
shopKeyword=keyword:shop [Product]
quoteKeyword=keyword:quote [Symbool hier invoeren]
localKeyword=keyword:zip [Je zip code]
keywordList=http://home.netscape.com/escapes/keywords
webmailKeyword=http://webmail.netscape.com
careerKeyword=keyword:[Je stad] careers
fallbackDefaultSearchURL=http://search.netscape.com/cgi-bin/search?charset=UTF-8&search=
otherSearchURL=http://home.netscape.com/bookmark/6_0/tsearch.html
#
# all.js
#
browser.startup.homepage=http://jeurl.com
browser.throbber.url=http://jeurl.com
browser.search.defaulturl=http://search.netscape.com/cgi-bin/search?search=

wallet.Server=http://www.mozilla.org/wallet/tables/
wallet.Samples=http://www.mozilla.org/wallet/samples/

#config.js
#
startup.homepage_override_url=http://jeurl.com
```

6. Compileer de broncode

Typ .configure vanuit de rootdirectory van de Mozilla broncodestructuur om Mozilla's compile-time opties te configureren. Typ ./configure --help voor een volledige lijst met compile-time opties. Wat je nog kunt verifiëren voordat je een compilatie voor een productie-omgeving uitvoert is dat de omgevingvoorkeuren (mail vs. no mail, calendar, ldap, enz.) zijn opgenomen en controleer of het crypto package is geactiveerd.

Ter vereenvoudiging, gebruik ik gewoonlijk een simpel build script zoals het volgende om Mozilla te configureren.

```
#!/bin/sh

MOZILLA_OFFICIAL=1
export MOZILLA_OFFICIAL
BUILD_OFFICIAL=1
export BUILD_OFFICIAL

./configure --with-x --disable-calendar --enable-crypto --with-system-nspr
--disable-debug --enable-extensions --enable-optimize
--without-system-zlib --without-system-jpeg --without-system-png
```

`--without-system-mng`

Zodra je de patches hebt toegepast, configuratiebestanden hebt aangepast, en opties hebt ingesteld, bouw je Mozilla simpelweg met gmake door het typen van ./gmake in de rootdirectory van je broncodestructuur, en maakt dan een tarball door het typen van gmake in de directory `xpinstall/packager`. De tarball komt terecht in de subdirectory dist direct onder de rootdirectory van je Mozilla broncodestructuur. Verplaats dat package gewoon naar `/usr/local`, pak het uit, voer het eenmaal uit vanaf de opdrachtregel als de gebruiker die het package bouwde (de opdracht is /usr/local/mozilla/mozilla), en je bent bijna klaar om te gaan browsen.

7. Configuratie na de installatie

Er zijn na de installatie van de browser verscheidene configuratiekeuzes te maken. Je kunt plugins toevoegen voor gebruikelijke webtechnologiën, zoals Java (java.sun.com (http://java.sun.com)), PDF printing (www.adobe.com), en Flash animaties (www.macromedia.com/software/flash (http://www.macromedia.com/software/flash)) en iets minder bekende, maar wel gave plug-ins, zoals (enigmail.mozdev.org (http://enigmail.mozdev.org)).

Kopieer voor het installeren van een plug-in de plug-in uit de broncodetoepassing (b.v. Acrobat5) naar de directory `/plugins` in Mozilla, of maak een symbolische link aan van de plug-in naar de plug-ins directory in de binaire structuur van Mozilla. Om bijvoorbeeld een Mozilla plug-in voor Adobe Acrobat 5 aan te maken met behulp van een symbolische link, typ je simpelweg ln -s /pad/naar/Acrobat5/Browsers/intellinux/nppdf.so /usr/local/mozilla/plugins/. Typ op vergelijkbare wijze ln -s /pad/naar/j2re1.4.0_02/plugin/i386/ns610/libjavaplugin_oji.so /usr/local/mozilla/plugins/ om een plug-in aan te maken van je Java runtime omgeving. Het is aan te bevelen dat je Sun's JDK 1.4.1 met Mozilla 1.0.1 en daarboven gebruikt. Er zijn incompatibiliteiten bekend met Mozilla en eerdere versies van Java.

Kijk op mozdev.org (http://mozdev.org) om aan Mozilla project plug-ins, zoals Enigmail, te komen, die in een interface voorziet voor het versleutelen en ontsleutelen van mail, of voor Protozilla, die voorziet in volledige parametrisatie (waaronder ondersteuning voor meerdere protocol en programmeertalen).