



Como Fazer: Montar compilador de C/C++ distribuído em FreeBSD, através do *distcc*

Francisco Alves Cabrita aka *include*

francisco@nortenet.pt

(versão 1.0 de 8 de Maio de 2006)

Resumo

Este documento tem em vista explicar como implementar um compilador de C/C++ distribuído através do **distcc** em FreeBSD. O uso do **distccd** é um demónio que nos permite definir um conjunto de servidores voluntários também a correr o **distccd** para que nas compilações, o código seja distribuído via TCP/IP para as máquinas definidas por nós para isso. A vantagem do uso de um sistema destes, é simplesmente, a de baixar os níveis de processamento da máquina onde estamos a instalar/compilar a aplicação.

1 O que é o distcc ?

O **distcc** é um programa para distribuir a compilação de código C ou C++ por varias máquinas numa rede. Este deve sempre gerar código resultado igual ao compilado na máquina local, é fácil de instalar, usar e torna a compilação cerca de 2.6 vezes mais rápida. Ao contrário de outros sistemas de compilação distribuídos, o **distcc**, não requer nenhuma partilha de sistemas de ficheiros por NFS ou outro, não requer ainda relógios sincronizados ou ainda que tenham as mesmas livrarias ou header files instalados.

1.1 Como funciona.

A compilação das aplicações é controlada por uma máquina cliente, que tipicamente é a do programador ou na qual a aplicação vai ser instalada. O **distcc** corre nesta, assim como o *make*, o pré-processador, o linkador, e todas as outras etapas do processo de construção/compilação. Qualquer número de máquinas voluntárias ajudam o cliente a compilar o programa/aplicação, correndo o compilador e o assembler como requerido. As máquina voluntárias correm o demónio **distccd** o qual escuta num *socket* de rede por ligações. O **distcc** envia completamente código fonte pré-processado através da rede para cada trabalho "*job*" assim o que se espera é máquinas definidas como voluntárias a correr o **distccd** e com um compilador instalado. O **distcc** está preparado para ser usado com a opção *-j* do *make* - compilador paralelo do GNU.

1.2 Onde se aplica ?

Temos vários destinos para a implementação de uma solução destas. No meu caso que em casa tenho uma pequena LAN composta por máquinas fracas (desde 486 33Mhz até ao PII 600Mhz) a correr FreeBSD, desta forma consigo um melhor desempenho/tempos a instalar aplicações nos servidores mais fracos.

Já em ambientes empresariais onde tudo que seja processado mais rápido tem maiores benefícios, podemos falar de uma empresa que desenvolve aplicações para Unix (ou derivados), onde o uso do **distcc** irá ajudar na tarefa ao compilar tudo distribuidamente.

1.3 Segurança.

O distcc deverá apenas ser usado em ambientes/Redes onde todas as máquinas e utilizadores são considerados de confiança.

Para aumentar minimamente a segurança deste demónio, os autores aconselham o uso do **inetd** ou **tcpwrappers**. Resumindo, o nível de segurança do **distcc** é equiparável ao antigo estilo de protocolos como **NFS** ou **RSH**.

PS: Está na minha óptica que nas futuras versões deste documento explorem este tópico, por enquanto peço paciência.

2 Instalar o distcc.

Tendo em conta possui uma coleção de portes e pacotes instalada e actualizada, deverá fazer os seguintes passos, independentemente se está na máquina do programador ou voluntária, este passo deverá ser igual em ambas:

```
#cd /usr/ports/devel/distcc/  
#make install clean
```

Até ao momento [05/05/2003] a versão do **distcc** é a 2.0.1, pode verificar através do comando:

```
#distcc --version  
distcc 2.0.1 i386-portbld-freebsd5.0 (protocol 1) (default port 3632)  
built May 2 2003 20:52:04
```

2.1 Configurar o distcc.

Para configurar o **distcc** no cliente basta definirmos algumas variáveis de ambiente como veremos de seguida:

```
#DISTCC_HOSTS='kitti prozak voltron lemon gluton localhost'  
#export DISTCC_HOSTS  
#CC=distcc  
#export distcc
```

Nota: Para que estas variáveis estejam presentes nos próximos logins, deveremos incluí-las em */etc/profile*.

kitti, *prozak*, *voltron*, *lemon* e *gluton* são nomes de máquinas "*hostnames*" definidos em */etc/hosts*, de lembrar que todas estas máquinas vão contribuir com o seu CPU nas próximas compilações.

```
#cat /etc/hosts  
#127.0.0.1      localhost.include.homeunix.net localhost  
#192.168.0.5   zarcon.include.homeunix.net zarcon  
#192.168.0.1   lemon.include.homeunix.net lemon  
#192.168.0.2   gluton.include.homeunix.net gluton  
#192.168.0.3   voltron.include.homeunix.net voltron  
#192.168.0.4   prozak.include.homeunix.net prozak
```

Para melhor distribuímos a compilação pelos voluntários, na variável **DISTCC_HOSTS** vamos ter em conta que estamos a listar as máquinas do CPU mais rápido para o menos rápido e acabando com o localhost, podemos ainda estrategicamente limitar o número de "trabalhos"/*jobs* que cada um pode correr e ainda a porta onde o **distccd** está a escutar da seguinte forma:

DISTCC_HOSTS=HOSTNAME/MAX_JOBS:PORTA

O uso de **localhost** ao invés do IP local da máquina "**127.0.0.1**", previne o **distcc** de criar uma ligação TCP ao demónio **localhost**, desta forma esta acção será directa, será mais lento mas útil para testes. Quanto à porta de ligação, por defeito será a **3632**.

2.2 Preparar o arranque do Servidor, **distccd**.

Agora já temos tudo apostos para preparar o arranque do **distccd**, para isso podemos inicia-lo de várias formas:

1. Manualmente:

```
# distccd --user nobody --daemon
```

2. Através dum rc.d script: */usr/local/etc/rc.d/distccd.sh*

```
#!/bin/sh
echo -n ' distccd'

case "$1" in
start)
    [ -x /usr/local/sbin/distccd ] && /usr/local/sbin/distccd \\\
--user nobody --daemon --pid-file /var/run/distccd.pid
    ;;
stop)
    [ -r /var/run/distccd.pid ] && kill -9 \\\
`cat /var/run/distccd.pid`
    ;;
*)
    echo "Usage: `basename $0` {start|stop}" >&2
    ;;
esac

exit 0
```

Depois de criar este script em todas as máquinas voluntárias, não esquecer de dar permissões de execução:

```
#chmod u+x distccd.sh
```

3. Através do *inetd.conf*

Para este último método, basta adicionar a seguinte linha a */etc/inetd.conf*

```
distccd stream tcp    nowait nobody /usr/local/sbin/distccd --inetd
```

3 Compilar/Instalar aplicações, verificação.

Se até aqui foi tudo muito fácil, bem, agora nada mais que os nossos fabulosos *make install* para darmos início ao uso do **distcc**, para isso podem por exemplo testar a instalação de uma aplicação tipo **GnuCash** que normalmente demora algum tempo :).

```
#cd /usr/ports/finance/gnucash
# make install clean
```

Entretanto se verificarmos as instruções executadas durante a instalação da aplicação podemos ver a seguinte saída no terminal: (*exemplo capturado*)

```
...
/bin/sh ../../libtool --mode=compile distcc -DHAVE_CONFIG_H -I. -I. \\\
-I../.. -I /usr/local/include -I/usr/X11R6/include -I../..../src/core-utils
...
```

Ou seja, o compilador executado foi o **distcc**, até aqui tudo indica que está operacional, no entanto podemos aprofundar mais um pouco o nosso teste, a fim de certificar se o código está a ser distribuído pelas máquinas voluntárias, executamos:

```
#sockstat -4 | grep 'distcc'
#root      distcc      13300 5  tcp4    192.168.70.4:49168  192.168.70.5:3632
#nobody    distccd     5867  3  tcp4    *:3632              *:*
```

Aqui finalmente concluímos que a máquina onde estamos a instalar o **GnuCash**, *prozak* (**192.168.70.4**) está a enviar código pré-compilado para a *zarcon* (**192.168.70.5**) na porta **3632** onde está a correr o demónio do **distcc**. Neste caso o **distcc** não teve necessidade de enviar mais nenhum "pedaço" de código para mais nenhum voluntário, pois este estava suficientemente livre no que diz respeito à percentagem de CPU usado. Divirtam-se.

4 Informação adicional

Pode encontrar a última versão deste CF em <http://npf.pt.freebsd.org>. Entretanto se tem algum comentário, crítica ou correcção a fazer sobre este CF, pode e deve fazê-lo para **francisco@nortenet.pt**. Obrigado.