



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

distcc, compilación distribuida entre redes (12236 lectures)

Per Celso González, [PerroVerd](http://mitago.net) (<http://mitago.net>)

Creado el 23/05/2002 12:24 modificado el 03/11/2002 00:14

distcc es un programa desarrollado por [Martin Pool](#)⁽¹⁾ que nos permite realizar compilaciones de código C distribuidas entre varios nodos de una red. *distcc* siempre genera los mismos resultados que una compilación local, pero con un incremento apreciable de velocidad. Ideal para una cluster de compilación. En este artículo explico como configurarlo, así como una pequeña introducción a la compilación y variables de entorno relacionadas.

Actualizado

Las nuevas versiones de distcc ya compilan C++, además un par de trucos para cierta Gentooza

```
MAKEOPTS="j9" CC=distcc CXX=distcc emerge paquete
```

Introducción

Tenemos una máquina principal, normalmente la estación de trabajo del programador, que realiza una llamada al programa cliente **distcc**, este mira que máquinas de la red van a colaborar en la compilación y transmite el código que deben compilar por la red. El código enviado es procesado por el programa servidor **distccd**. Una vez que está listo el código a compilar se devuelve a la máquina principal.

La principal ventaja de este programa es que las máquinas colaboradoras no tienen que tener las mismas versiones de librerías, ficheros de cabeceras o sistemas de ficheros compartidos

En los ejemplos voy a usar 2 máquinas denominadas *k6* (un k6-2 350 con 384 Mb de RAM) y *pentium* (un Pentium 100 con 48 Mb de RAM), *k6* será la máquina principal y *pentium* será una máquina colaboradora.

Un poco de teoría de compilación

La compilación de un programa en C consta de 4 fases

- Preprocesamiento de los ficheros fuentes (.c) y de cabeceras (.h) lo que nos origina un fichero preprocesado (.i)
- Compilación de un fichero preprocesado (.i) a instrucciones de ensamblador (.s)
- Ensamblado del fichero (.s) para generar un fichero de objeto (.o)
- Enlazado del fichero objeto (.o) para formar un ejecutable o una librería.

distcc únicamente transmite ficheros (.i) y (.s) por la red, de esta forma no necesita usar ficheros de cabeceras ni librerías específicas. Además la primera fase es bastante ligera

Instalación

Lo primero que tenemos que hacer es bajar el código fuente y compilarlo. La página desde donde podemos bajar el código es [esta](#)⁽²⁾, aunque también podemos usar el código CVS del servidor de [samba](#)⁽³⁾ (modulo distcc)

- Descomprimos el tar.gz con `tar xzvf distcc-x.x.tar.gz`
- Vamos al directorio `distcc-x.x/src`



- Ejecutamos `make` y después como `root` `make install` (Nota. Si no queremos compilar la documentación (requiere `tetex`, `gs` y `latex2html`) podemos editar el archivo `distcc-x.x/src/GNUMakefile` y comentar con un `#` la línea de docs)
- Listo, el cliente lo encontraremos en `/usr/local/bin/distcc` y el servidor en `/usr/local/sbin/distccd`

Ejecución del servidor

El servidor se tiene que ejecutar en todas las máquinas que van a participar en la compilación (principal y colaboradoras). Para ejecutarlo basta con escribir desde una consola como `root` `/usr/local/sbin/distccd&`

Si hacemos un `ps aux` veremos si se está ejecutando, también veremos que tenemos abierto el puerto TCP 4200. Ver notas de seguridad más adelante

Variables de entorno

El programa se controla gracias a 3 variables de entorno, siendo `DISTCC_HOSTS` la más importante, estas variables únicamente son necesarias en el ordenador principal.

- `DISTCC_HOSTS`

Esta variable nos sirve para indicar los hosts que van a colaborar en la compilación, en nuestro caso vamos a utilizar los hosts `k6` y `pentium`, por lo que en una consola bash escribiríamos esto

```
export DISTCC_HOSTS='k6 pentium'
```

Un par de notas importantes, el orden influye mucho en la eficiencia que obtengamos, las reglas generales son las siguientes: utilizar `localhost` como el primer nombre de host, a continuación el resto de colaboradores ordenados por velocidad, los más rápidos primero y los más lentos después.

- `DISTCC_LOG`

Esta variable nos permite indicar un fichero de logs, escribiendo esto en la consola conseguiremos que nos guarde los logs de la ejecución en el fichero `/var/log/distcc.log`

```
export DISTCC_LOG=/var/log/distcc.log
```

- `DISTCC_VERBOSE`

Asignando el valor 1 a esta variable conseguiremos que el log sea mucho más extenso, normalmente esto solo es necesario a efectos de depuración

Uso de distcc

La sintaxis de `distcc` es muy sencilla, por ejemplo para compilar el clásico Hola mundo normalmente haríamos

```
gcc -o hello.c
```

Ahora lo único que tenemos que hacer es indicar que queremos usar `distcc` y que el compilador que usaremos será el `gcc`

```
distcc gcc -o hello.c
```

Una cosa que nos puede ayudar mucho es la variable de entorno `CC`, esta variable se mucho en los ficheros `Makefile` para indicar que compilador es el que se va a usar facilitando la portabilidad. Por ejemplo, si tenemos un programa `X` con un `Makefile` estandar y queremos utilizar el compilador de Intel lo único que tenemos que hacer es hacer la llamada a `make` utilizando ese valor.

```
make CC='icc'
```

Si queremos compilar algo utilizando la versión 3.1 del `gcc` podemos hacer

```
make CC='gcc-3.1' (o también make CC='-V 3.1')
```

En nuestro caso, para poder usar `distcc` indicaremos ese valor en el `make`, aunque además añadiremos otro parámetro que nos sirve para indicar el número de instancias simultaneas de `make` que se van a utilizar (normalmente un número ligeramente superior al número de hosts colaboradores)

```
make -j3 CC='distcc gcc'
```



Makefiles

Normalmente con lo que he explicado arriba debería bastarnos para compilar cualquier cosa, pero seguramente nos encontraremos con un problema. Muchos de los Makefiles llevan definida dentro la variable CC, en este caso aunque nosotros le pasemos la variable, su valor es machacado por la que contiene el fichero.

Todas las posibles soluciones pasan por editar el Makefile, un ejemplo puede ser el Makefile del kernel. En el encontramos 2 línea similares a estas

```
CC = $(CROSS_COMPILE)gcc y
export ... CC ...
```

Las soluciones pueden ser modificar el valor de CC directamente en el fichero (CC = 'distcc \$(CROSS_COMPILE)gcc') o bien eliminar el export e incluir el valor de CC en la llamada a make

Tests

Mi test consiste en la compilación del kernel utilizando las dos máquina indicadas. Los resultados son los siguientes y se puede apreciar un incremento de velocidad del 18%

```
Usando distcc
real    16m58.298s
user    15m4.500s
sys     1m26.350s
```

```
Compilación normal
real    20m50.873s
user    18m17.220s
sys     1m25.750s
```

En la documentación del proyecto podemos encontrar un amplio número de tests adicionales

Consideraciones de seguridad

Este programa no está diseñado para ser seguro, tenemos un puerto TCP abierto y un atacante puede ejecutar código arbitrario con los permisos del usuario de distccd, por otro lado también tenemos que el código preprocesado está viajando por la red sin ningún tipo de encriptación. Tal como dice el propio autor "**distccd solo debería ser usado en redes donde todos las máquinas y usuarios son de confianza**".

Se puede aumentar la seguridad usando mecanismos como *inetd* o *tcpwrappers*, aunque el nivel de seguridad sigue siendo bajo.

La otra opción es usar redireccionamiento de puertos con ssh, aunque esto supone una carga mayor de las máquinas y un retardo adicional en la compilación

Conclusiones

Un programa excelente y de uso obligado si necesitas compilar con frecuencia. Según vaya mejorando (solo es la versión 0.2), especialmente en temas de seguridad, un programa ideal para instalar en clases de prácticas o en grupos de trabajo, o para aprovechar esos ordenadores que tenemos amontonados por la habitación :)

Lista de enlaces de este artículo:

1. <http://sourcefrog.net/mbp/>
2. <http://samba.org/~mbp/distcc/>
3. <http://cvs.samba.org>

E-mail del autor: celso _ARROBA_ mitago.net

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1327>