Table of Contents

distcc Pilot Service

Announcements

Introduction

Welcome to the distcc pilot service TWiki page. The purpose of this page is to provide information to users involved in the distcc pilot service, and to new users who may be interested.

Quickstart

Main idea is that people can submit `distcc` jobs from their workstation, laptop or build host. To do this, you'll need to install the following packages (as root):

SLC6 64-bit:

- `yum install distcc cern-lxdistcc-hosts`

SLC5 64-bit:

- `yum install distcc cern-lxdistcc-hosts`

Quattor: see in the FAQ

Start using the `lxdistcc` cluster:

1. `kinit user@CERN.CH` # this gets you Kerberos credentials, if you don't have any
2. `make CC distcc -j16` # compile away!

(For people already familiar with `distcc`: no need to set up `DISTCC_HOSTS` variable (in fact it is advised to unset `DISTCC_HOSTS`) because hostname are taken from `/etc/distcc/hosts`.)

What is `distcc`?

Most visitors to this page will already know about distcc and its advantages, but for those that don't, here's a brief description.

distcc is a tool for speeding up software builds by distributing compilition jobs across several machines on a network. It works with C, C++, Objective C, and Objective C++, and is usually much faster than a local build.

For further information, please refer to the documents in the Useful Links section.

CERN Improvements

- **GSSAPI Authentication**: a prerequisite for a shared resource, currently implemented with Kerberos V.
- **Whitelist / Blacklist**: resource control to protect the service from abuse.
- **Log Timestamps**: to aid troubleshooting.

We're working with `distcc` maintainers (Google) to include these improvements in future upstream versions.

Pilot Goals

- Test CERN improvements in an operational environment.
- Obtain user feedback to better adapt to experiment needs.
- Evolve the service where appropriate.
- Migrate existing distcc users.
- Invite new interested parties.

Current Cluster Configuration

Quattorized server nodes

- 8 x 8 = 16 E5410 cores SLC6 / 64-bit (pre-production).
- 8 x 8 = 96 E5410 cores SLC5 / 64-bit (production).

`lxdistcc` is a shared service for all experiments and individual users.

Future (goals towards a real service)

- Upstream acceptance of CERN patches
- Usage Statistics
- Remedy support line
- LEMON monitoring
- User prioritization (based on GSSAPI auth)

Useful Links

Official `distcc` home page. ☑
CERN `distcc` GSSAPI user documentation. ☑
Peter Kelemen's AA meeting presentation. ☑

CERN IP networks list in LANDB ☑
`lxdistcc` in LEMON ☑
`lxdistcc` in CERN Service Database ☑

Mailing List

linux-distcc@cernNOSPAMPLEASE.ch

Please subscribe to this mailing list if you are interested in following news about the `lxdistcc` cluster. We are building a community of `distcc` users at CERN and we would like this mailing list to be the first place to ask questions and discuss issues (after reading the FAQ, of course).

Frequently Asked Questions

What is the `lxdistcc` cluster?

Managed servers in the CERN Computer Centre that run CERN-modified `distcc` daemon (the most prominent is GSSAPI authentication).

Who can use the `lxdistcc` cluster?

Every CERN user from all CERN machines. You have to have a valid Kerberos V credential and your IP address must be within CERN.

What software is required in order to use the `lxdistcc` cluster?

You need the build environment of your software project (obviously) and the `distcc` client. It is part of the Scientific Linux CERN distribution (package name `distcc`).

Is the `lxdistcc` cluster any different than my own `distcc` cluster?

Yes, `lxdistcc` servers use GSSAPI authentication to distinguish among users. It means that you will need the SLC-distributed `distcc` client in order to be able to use it.

Authenticated connections are initiated to hosts that have host definitions of the form `HOST,gssapi`. This way it is possible to talk to authenticated and non-authenticated at the same time.

What platforms are supported?

`lxdistcc` cluster comes in many flavours, out of which currently SLC6 64-bit and SLC5 64-bit are supported. Please note that it is strongly recommended to have the `distcc` client platform matched to the cluster you wish to use.

What compilers are supported?

The compile nodes in `lxdistcc` cluster allow the following system compilers to be used:

- `/usr/bin/cc`
- `/usr/bin/c++`
- `/usr/bin/c89`
- `/usr/bin/c99`
- `/usr/bin/gcc`
- `/usr/bin/g++`

In addition, several of the LCG AA-provided compilers have been made available. At the time of writing (July 2013), the supported compilers are:

- SLC5:
  - `/usr/bin/lcg-{gcc,g++,c++}-4.3.2`
  - `/usr/bin/lcg-{gcc,g++,c++}-4.3.5`
  - `/usr/bin/lcg-{gcc,g++,c++}-4.3.6`

- ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.5.2`
- ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.6.2`
- ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.6.3`
- ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.7.2`
- ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.8.0`
- ◆ `/usr/bin/lcg-{clang,clang++}-3.2`
- SLC6:
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.3.5`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.3.6`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.5.3`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.6.2`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.6.3`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.6.3`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.7.2`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.8.0`
  - ◆ `/usr/bin/lcg-{gcc,g++,c++}-4.8.1`
  - ◆ `/usr/bin/lcg-{clang,clang++}-3.2`

How does `distcc` work?

`distcc` is implemented as a wrapper around `gcc`. It invokes `gcc` to do the preprocessing on the client side, then sends the resulting compilation unit off to a `distcc` server (together with the options passed to `gcc`), which compiles it and sends the resulting object file back (or error messages, if any). Then linking again takes place on the client.

This all means that your software project should be able to build in parallel. The easiest way to verify this is `make -j4`, this will use four parallel `make` processes to build. If it fails (i.e. your software doesn't compile), that means your software build procedure is not parallelizable due to various dependencies (in which case you will have to fix it first before you can benefit from `distcc`). If it succeeds, you have a good chance that `distcc` will be beneficial for you, but it is not yet a guarantee by itself. Correctness of parallel builds hinges on correct build dependencies of your software project.

How fast is `distcc`?

It is a tough question to answer. A lot depends on how the build process of your software project is organized. In short, the more parallel it is, the better `distcc` can speed it up. By parallel we mean different parts of your software that can be compiled independently. Also, software build procedures usually include non-compilation tasks as well, like documentation generation etc. which obviously cannot benefit from `distcc`.

With well-behaving build environments approx. 80% of perfect linear speedup can usually be observed.

Can I use the CERN `distcc` client with any other `distcc` cluster?

Yes.

Quattor-managed nodes can have the client too?

Yes!

For SLC6 64-bit:

```
include services/distcc/config;
```

What compilers are supported?   4

For SLC5 64-bit:

```
include services/distcc/config;
```

How about `pump` mode?

Currently `pump` mode is **not** supported.
SLC5/6: No testing had been performed yet.

People behind `lxdistcc`

Linux.Support@cernNOSPAMPLEASE.ch