Table of Contents

Installing the LHCb Software on Ubuntu (and Debian)

Introduction

"we have no plans to create a traditional desktop product for the consumer market in the foreseeable future." -- redhat⌐, 2008/04/16. This leaves us with the real stuff: Debian⌐, or rather Ubuntu.

Packages of CERN software residing in the CERN package repository can be found here⌐.

System Installation and Tuning

In the following sections it is explained how to install Ubuntu and tune the installation in useful ways.

Getting Ubuntu

- Everything below works on 9.04 (and can upgrade flawlessly; but you do need to repeat the AFS step below).
- The smoothest way to install is simply to use the "Desktop" edition
- For LVM-s and funny hardware, choose the "Alternate" installation image. If you like KDE, you can do `sudo aptitude install kubuntu-desktop`, even if you didn't use Kubuntu install media
- When setting mirror for package sources, take care not to push "Select best mirror" button within CERN network, since it triggers ICMP traffic exceeding the CERN sec. policy threshold.
- make sure, that your system is up-to-date by typing: `sudo aptitude update` and `sudo aptitude dist-upgrade` or `sudo aptitude safe-upgrade`

Installing "NICE" Alerter (RSS notifier)

- `sudo aptitude install liferea` and add a new feed `http://cernalerts.web.cern.ch/cernalerts/alerts.aspx`. Adding the feed reader in System->Settings->Sessions will launch it upon login.

Installing ROOT (properly)

Note: this is not strictly needed because we distribute all the required special libraries in our software kits.

- ROOT has become part of Debian as of Lenny. Look for packages root-system (~-common and all other libraries, modules, plugins etc.)

Installing CLHEP and others (properly)

Note: this is not strictly needed because we distribute all the required special libraries in our software kits.

- CLHEP has become part of Debian as of Lenny. Look for packages DebianScience/Physics⌐ homepage

Installing required packages

- `sudo aptitude install cvs build-essential krb5-clients openafs-client openafs-modules-source module-assistant openafs-krb5 krb5-auth-dialog libmotif3 libmotif-dev xnest`. Our AFS Cell and Kerberos server is `cern.ch` # => generate /afs dynamic =>

yes (but it will show up in the configuration tool...). If you think you made an error don't worry, you can always run `sudo dpkg-reconfigure openafs-client` later on. Optionally you can `sudo cp /afs/cern.ch/common/CellServDB /etc/openafs/`

- ATTENTION! the following step will be needed after each kernel update: `sudo module-assistant auto-install openafs`
- restart OpenAFS: `sudo /etc/init.d/openafs-client force-start`
- and maybe you might also want to install a nice development environment (apart from emacs and vim `sudo aptitude install eclipse eclipse-cdt svn meld kompare maxima koctave libgnome-vfs-common sshfs openjdk-6-jre gnash gnome-do gnome-do-plugin* valgrind alleyoop kcachegrind` etc. feel free to browse "System->Administration->Synaptic" and "Applications->Add/Remove...". Optionally Oracle for Ubuntu⬚ is available for local installation (though the LCG on AFS also works fine). I suggest using Eclipse with Open Java 6, it is much faster than gcj (look inside the file `/etc/eclipse/java_home` and add the line `/usr/lib/jvm/java-6-openjdk` if needed).

CERN issued multimedia keyboards

- `sudo aptitude install keytouch keytouch-data keytouch-editor` helps to bind special keys.

Using CMT

- after having installed everything above, you can authenticate yourself to access your AFS directories: `klog.afs cernUsername@cern.ch`
- save your current PATH entries to SAVEPATH, because CMT scripts will overwrite it...
- now you can source the appropriate scripts (`CMT.[c]sh` likes to set `gcc433-5ubuntu4` in CMTCONFIG), so take care to set CMTCONFIG before and after CMT.[c]sh:

`. /afs/cern.ch/lhcb/scripts/lhcbsetup.sh`; export CMTCONFIG=x86_64-slc5-gcc43-opt; `. /afs/cern.ch/lhcb/scripts/CMT.sh`; export CMTCONFIG=x86_64-slc5-gcc43-opt

CVS access via SSH

- after having sourced your project's setup, launch `eclipse&` from the given prompt. Then choose File->New Project->Standard Make C++ Project and navigate to your project directory. If you put a stub Makefile just below the project version directory and point eclipse project to this directory, then Build and Clean will also automagically work, along with the interactive debugging view.
- an example stub Makefile:

```
all:
    $(MAKE)  all -C cmt
# Any specified target is redirected to the cmt directory for
# building.
%:
    $(MAKE)  $* -C cmt
```

Vim with LHCb coding conventions

You might want to add the following lines to your `~/.vimrc`:

```
set showfulltag
set softtabstop=2
set tabstop=2
set shiftwidth=2
set expandtab
```

and optionally some other useful things:

```
syntax on
set number
```

Accessing CASTOR

You might also like to try to also mount CASTOR via fuse CastorFS⬚.

Printing

CERN documentation⬚

Select Administration-->Printing and select add new printer. Choose Network: Unix Printer (LPD/LPR Host or Printer).

- Enter the Host name as the full printer DNS name BUT without the letter "B" and ended by ".print.cern.ch" ex. "2-cor" must be entered as `2-cor.print.cern.ch`
- Enter the Queue name, using the normal DNS name BUT also without the letter B ex. "B2-cor" must be entered as `2-cor`
- Select the correct manufacturer and printer model or family ex. "HP" and `HP LaserJet 8150 Series Postscript (recommended)`, click on "Add" and "Continue"

Backup, restore

To backup your data, you might like flyback⬚. For your package selection, type `sudo dpkg --get-selections > /tmp/dpkglist.txt` and after you have made a clean install (for whatever reason), you can type `sudo dpkg --set-selections < /tmp/dpkglist.txt`, then `sudo apt-get -y update` and finally `sudo apt-get dselect-upgrade` to restore your package selection.

Installing and Using LHCb Software

LHCb precompiled software can work on Ubuntu, but a few special actions are needed. (the following instructions have been tested on (K)Ubuntu 12.04)

First follow the installation instructions at SoftwareInstallation, but set `CMTCONFIG` to `x86_64-slc5-gcc46-opt`, if the application you want to install has been built on gcc 4.6.

This is usually enough to be able to run applications, but if you want to compile, you have to follow the following steps:

- remove the directory `$MYSITEROOT/lcg/external/gcc`
  gcc 4.3 cannot understand some system headers present on recent platforms (gcc >= 4.4), so it is better to use the system compiler
- install the version of gccxml from Ubuntu (the version of gccxml we distribute does not understand the headers too)
  ```
  sudo apt-get install gccxml
  ```
- To build LHCb software, Install other needed dependencies:
  ```
  sudo apt-get install svn gccxml procmail
  ```
- Create a "gmake" link to make:
  ```
  sudo ln -s /usr/bin/make /usr/bin/gmake
  ```
- For SLC6 configurations, Make sure there is a libcrypto and libssl with the proper name:

  ```
  sudo ln -s /lib/x86_64-linux-gnu/libssl.so.1.0.0 /lib/x86_64-linux-gnu/libssl.so.10
  ```

```
sudo ln -s /lib/x86_64-linux-gnu/libcrypto.so.1.0.0 /lib/x86_64-linux-gnu/libcrypto.s
```

- create the directory `$MYSITEROOT/cmtusercontext` and create inside it a file called `requirements` with this content:

```
macro_remove cpp_name "lcg-"
macro_remove cpp_name "-$(gcc_config_version)"
macro GCCXML_home "/usr"
```
- create the file `$MYSITEROOT/MyLbLogin.sh`, in which you should put something like this:

```
msr=/path/to/my/siteroot
export CMTUSERCONTEXT=$msr/cmtusercontext
. $msr/LbLogin.sh -c x86_64-slc5-gcc46-opt
```

  where, of course, you replaced `/path/to/my/siteroot` with the path you used to set `$MYSITEROOT`, and the argument to LbLogin with the CMTCONFIG you actually used.
- if your system language is not English, add this line to the created `MyLbLogin.sh`:

```
export LANG=C
```

Now, to set the environment up to work, just *source* the script `MyLbLogin.sh` instead of `LbLogin.sh`.

This topic: LHCb > LHCbSLCOnDebian
Topic revision: r41 - 2013-05-15 - BenjaminCouturier