Quantian: A single-system image scientific cluster computing environment

Dirk Eddelbuettel, Ph.D.

B of A, and Debian

edd@debian.org

Presentation at the *Extreme Linux SIG* at USENIX 2004 in Boston, July 2, 2004

Introduction

- Quantian is a directly bootable and self-configuring Linux sytem that runs from a compressed dvd image.
- Quantian offers zero-configuration cluster computing using openMosix.
- Quantian can boot ’thin clients’ directly via PXE in an ’openmosixterminalserver’ setting.
- Quantian contains around 1gb of additional ’quantitative’ software: scientific, numerical, statistical, engineering, ...
- Quantian also contains tools of general usefulness such as editors, programming languages, a very complete latex suite, two ’office’ suites, networking tools and multimedia apps.

Family tree overview

- Quantian is based on clusterKnoppix, which extends Knoppix with an openMosix-enabled kernel and applications (chpox, gomd, tyd, ...), kernel modules and security patches.
- ClusterKnoppix extends Knoppix, an impressive ’linux on a cdrom’ system which puts 2.3gb of software onto a cdrom along with the very best auto-detection and configuration.
- Knoppix is based on Debian, a Linux distribution containing over 6000 source packages available for 10 architectures (such as i386, alpha, ia64, amd64, sparc or s390) produced by hundreds of individuals from across the globe.

Family tree: Debian

- ’Linux the Linux way’: made by volunteers (some now with full-time backing) from across the globe.
- Focus on very high technical standards with rigorous policy and reference documents.
- Open: if you don’t like a given aspect, you can join the project and work on improving it.
- Rather large scope:
  - several hundred people,
  - about a dozen different computer architectures, and
  - over twelve thousand *binary* packages based on around six thousand *source* packages
- Used as a basis for derivative distributions such as Progeny, Corel/Xandros, Lindows/Linspire, Knoppix and all its derivatives, work by European governments, . . .

Family tree: Knoppix

- Usable on any recent desktop or laptop — lowers barriers to entry for Linux and Unix technology via ready-to-use system;
- contains lots of utilities, tools, games, eyecandy, and is *e.g.* usable for system recovery and forensics due to a large number of included tools and utilities;
- permits one to try Linux risk-free as no information is written to the hard disk, and enables to try Linux on new hardware to reliably test its compatibility;
- provides a terminalserver mode allowing other netboot-capable machines to be initialized and booted using the PXE protocol;
- provides a ‘persistent home’ mode where data can be written to USB storage devices (or disk partitions) to preserve state;
- makes for easier installation of ’permanent system via `knoppix-installer`.

Family tree: clusterKnoppix

- Adds openMosix kernel patch, and specific tools such as openmosixview, gomd, tyd to create ’Knoppix for clusters’;
- extends terminal server mode for openMosix permitting clients to boot via the network and join the cluster with zero configuration;
- operates openMosix in autodiscovery mode so that new nodes automatically join the cluster;
- creates setup where every node has root access to every other node via ssh;
- provides Mosix File System (MFS/DFSA) support which enables all nodes to see each others files;
- also adds a couple of networking tools, in particular for wireless operations.

Timeline

- 0.1 (March 2003): Initial version presented at DSC 2003 conference.
- 0.2 (May 2003): Now based on Knoppix 3.2.
- 0.3 (June 2003): Switched to using clusterKnoppix which added openMosix clustering support.
- 0.3.9.1 to 0.3.9.3 (September 2003): Updated clusterKnoppix; final version re-released as 0.4 in October 2003.
- 0.4.9.1 to 0.4.9.6 (October 2003 to March 2004): Based on Knoppix 3.3, final version also released as 0.5.
- 0.5.9.1 and 0.5.9.2 (June 2004): Now based on Knoppix 3.4, released as 'kitchen sink' versions > 1gb for bootable DVDs or booting from hard disk.

Motivation

- **Computing clusters** speed up embarrassingly parallel tasks.
- **Computer labs** by enabling temporary use of a computing environment booted off a dvd, and netbooting other machines.
- **Students and co-workers** where distributing DVDs enables working in identical environments with minimal administration.
- **Convenience** of not having to chase down new software releases, and to configure and installing it, and
- **Easier installation** of a ’normal’ workstation by booting off Quantian, and installation that system to hard disk thus getting a head start with 3.6gb of configured software.
- **Travel** At conferences or other campuses, convenient to carry on research in a familiar environment.

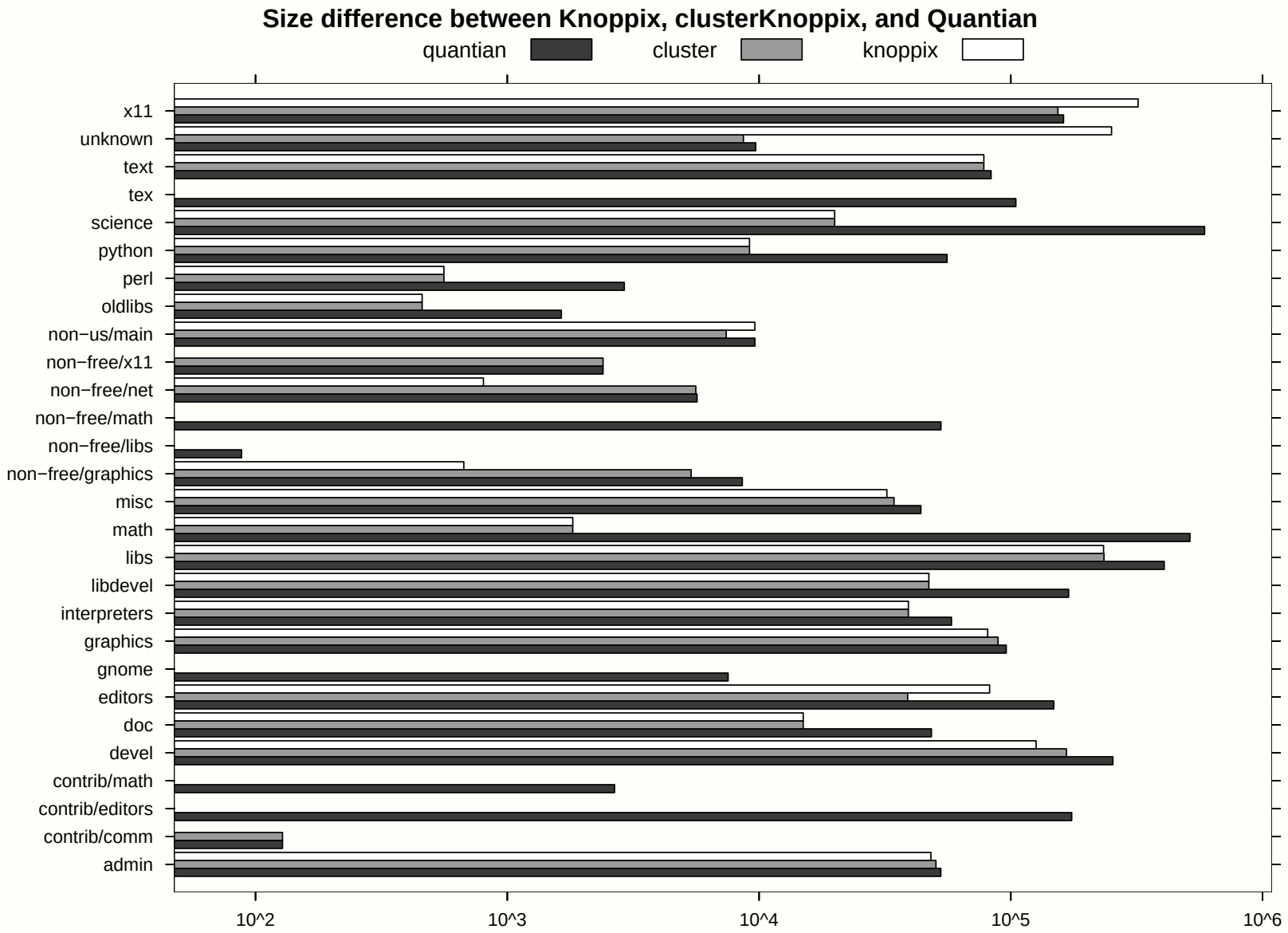
So what is included?

- **Mathematics:** Computer algebra systems Maxima, Pari/GP, GAP, GiNaC, YaCaS and Axiom; matrix-oriented languages Octave (with packages octave-forge, matwrap, octave-epstk), Yorick and Scilab, and the TeXmacs front-end.
- **Statistics:** GNU R (with numerous packages from CRAN, BioConductor, Rmetrics and other archives, as well as Ggobi and ESS tools), Xlispstat, Gretl, PSPP, X12A.
- **Bioinformatics:** BioConductor, BioPython, BioPerl and tools like emboss and blast2.
- **Physics:** CERN tools like Cernlib, Geant, PAW/PAW++; Scientific and Numeric Python and the GNU GSL libraries.
- **Visualization and graphics:** OpenDX, Mayavi, Gnuplot, Grace, Gri, plotutils, xfig.

So what is included (cont.)?

- **Finance:** Software from the Rmetrics project and the QuantLib libraries.
- **Programming languages:** C, C++, Fortran, Java, Perl, Python, PHP, Ruby, Lua, Tcl, Awk, A+.
- **Editors:** XEmacs, Vim, jed, joe, kate, nedit, zile.
- **Scientific Publishing:** Extended LaTeX support with several frontends (xemacs, kile, lyx) and numerous extensions.
- **Office software:** OpenOffice.org, KOffice, Gnumeric, and tools like the Gimp.
- **Networking:** ethereal, portmap, netcat, ethercap, bittorent, nmap, squid and a host of wireless tools and drivers.
- **General tools:** Apache, MySQL, PHP, and more.

So what is included (cont.)?



Cluster computing: openMosix

- Easiest way to distribute computing load, esp. for ’embarrassingly parallel’ tasks, as the kernel schedules tasks across the cluster.
- Since release 0.3, clusterKnoppix has added a kernel with the openMosix patch as well as a set of openMosix utilities such as openmosixview, gomd, chpox, tyd.
- As a result, “instant cluster computing” is possible based on a single dvd or iso image:
  1. boot one master instance from the dvd or hard disk,
  2. enable ’openmosixterminalserver’ from the menu after answering a few simple configuration questions,
  3. boot 1, 2, . . . ’slave’ nodes via the PXE protocol (enabled in most recent computers) from the master, and
  4. enjoy openMosix on the cluster.

Cluster computing: openMosix (cont.)

- clusterKnoppix autoconfigures and autodiscovers the nodes and enables ssh access between them.
- Ian Latter's CHAOS/tyd projects can address some of the security aspects by overlaying a VPN allowing for private clusters on top of public networks.
- openMosix is ideal for stand-alone programs such as 'old fashioned' C++ or Fortran apps that 'just run'.
- Mix-and-match with clusterKnoppix is easiest – though any indentical kernel and openMosix version could join.
- In general, any program without shared memory use, or threads, will migrate (c.f. the 'work smoothly' page at the openMosix wiki).

Cluster computing: Beowulf

- openMosix takes existing programs and moves them around nodes in the cluster to achieve optimal load across all nodes in the cluster – this requires no alteration to algorithm, or new programming.
- Beowulf cluster, on the other, require the programmer to use message-passing interfaces such as MPICH or PVM to communicate across nodes – this may require a sizable amount of new programming.
- Quantian includes several Beowulf tools and libraries:
  - Lam (Local Area Multicomputer) MPI libraries and run-time;
  - Mpich MPI libraries and run-time;
  - Pvm (Parallel Virtual Machine) libraries and run-time;
  - Sprng (Scalable Parallel Random Number Generator) RNG.

Cluster example: SNOW

- Tierney et al. introduced the 'Simple Network of Workstations' (SNOW) for R, similar to the 'Cluster of Workstations' (COW) concept for Scientific Python.
- Snow takes care of all inter-node communications allowing the user to concentrate on higher-level abstractions rather then 'the plumbing'.
- Snow can use sockets, pvm or mpi to communicate, and includes support for sprng to ensure RNG streams are suitable for parallel computations.
- Snow employs the existing CRAN packages rmpi, rpvm, rsprng to provide the base functionality, and ties them together to provide truly easy access to high-level parallel (statistical) computing.

Cluster example: SNOW (cont.)

- We illustrate Snow with a simple extended bootstrap example provided by Luke Tierney.
- The example is based on code from the boot package providing functions and datasets from the classic Davison & Hinkley (1997) monograph; it is for a regression example based on the 'nuclear' data set for costs of light water reactors.
- The basic, non-parallelised bootstrap, uses code such as

```
library(boot)
data(nuclear)
[...]
nuke.boot <-
  boot(nuke.data, nuke.fun, R=nbBootstraps,
       m=1, fit.pred=new.fit, x.pred=new.data)
```

where `nuke.boot` is the returned bootstrap object.

Cluster example: SNOW (cont.)

- This can very be executed very easily in parallel:

```
library(rsprng)
library(snow)
[...]
clusterEvalQ(cl, z<-library(boot))
[...]
cl <- makeCluster(nbClusters, "MPI")
clusterSetupSPRNG(cl)
[...]
cl.nuke.boot <-
  clusterCall(cl,boot,nuke.data, nuke.fun,
     R=round(nbBootstraps/nbClusters),
     m=1, fit.pred=new.fit, x.pred=new.data)))
```

where `cl.nuke.boot` is a list containing the per-node returned bootstrap objects.

Cluster example: SNOW (cont.)

- Many modern statistical computing applications (e.g. MCMC, bootstraps or boosting) require simulations. As well, nonlinear optimization or sensitivity analysis must often be re-run with slight variations in the starting parameters.
- Such 'embarassingly parallel' problems can profit immensely from a cluster: in the simplest cases, M sequential runs could be executed in parallel on M nodes.
- The example showed that we can easily embed parallelism in the algorithms at a high-level in R without requiring a new tool or language.
- Best of all, we can launch (approximately) N virtual cluster nodes (using either PVM or MPI) with Snow . . . and openMosix takes care of moving these around for us – freeing us from having to coordinate PVM or MPI across nodes.

Current issues

- **Size:** A constant problem:
  - Users like to see more and more software added.
  - We would also like to add more if not all of the available documentation.
  - Lastly, Open Source software has a tendency to grow rapidly in scope and size.

  and DVDs appear to be the best answer with harddisk-based booting as an alternative.
- **Selection:** Currently no real mechanism to survey users for additional software needs and configurations.
- **Security:** Ongoing work by Latter et al on better security in open clusters based on OpenSWAN-based VPN layers over the (possibly public) openMosix communications layer.

Conclusion

- Quantian offers a complete scientific computing environment.
- Based on Knoppix, it ’just works’ on most (i386) platforms.
- Based on clusterKnoppix, it also offers openMosix clustering with zero configuration and can boot ’fat clients’ in a terminal server function.
- Quantian contains over 3.5gb of relevant software drawn from math, stats, sciences, engineering as well as general programming tools and environments.
- Quantian offers single system image clustering and/or the ability to do Beowulf-style directly controlled communication to other nodes.

Acknowledgements

- Tony Rossini, Luke Tierney and Michael (Na) Li for helping me with Snow, and Kai Hendry and Elijah Wright for help in getting sprng built
- Ian Latter, Bruce Knox, Mathias Rechenburg and others from the openMosix community for patiently answering my questions;
- and, of course, Wim Vandersmissen for clusterKnoppix; and Klaus Knopper, Fabian Franz, Christian Perrier and others for Knoppix'
- the Debian Project for Debian, and, last but not least, the upstream authors for making their code freely available;
- and finally Lisa, Anna and Julia for allowing me my weekend and evening computing habit.

References

Moshe Bar et al., *openMosix*, 2004. `http://www.openmosix.org`

A.C. Davison and D.V. Hinkley, *Bootstrap Methods and Their Applications*, Cambridge University Press, 1997.

Dirk Eddelbuettel, *Quantian*, 2004. `htpp://dirk.eddelbuettel.com/quantian.html`

Dirk Eddelbuettel, *Quantian: A Scientific Computing Environment*, Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), ISSN 1609-395X.

Klaus Knopper, *Knoppix*, 2004. `http://www.knopper.net/knoppix/index-en.html`

Ian Latter, *Security and openMosix: Securely deploying SSI cluster technology over untrusred networking infrastructure*, manuscript, December 2003, Macquarie University. `http://itsecurity.mq.edu.au/papers/WhitePaper-SecurityandopenMosix.pdf`

Anthony Rossini, Luke Tierney and Na Li, *Simple Parallel Statistical Computing in R*, UW Biostatistics Working Paper Series, Number 193, 2003. `http://www.bepress.com/uwbiostat/paper193`

Wim Vandersmissen, *ClusterKnoppix*, 2004. `http://bofh.be/clusterknoppix`