Deploying a Network of GNU/Linux Clusters with Rocks

Arto Teräs <arto.teras@csc.fi>

Free and Open Source Software Developers' European Meeting

Brussels, February 26th, 2005

Contents

- Rocks Cluster Distribution
- Task: The Finnish Material Sciences Grid
- Installing a Rocks Master Server
- Customizing Rocks
- Experiences in deploying the systems
- Rocks pros and cons
- Short comparison with Debian FAI
- Questions

NPACI Rocks Cluster Distribution

- **Cluster oriented GNU/Linux distribution**
- **Main developers in the San Diego Supercomputing Center (U.S.A.)**
- **Base packages, installer components and kernel from Red Hat Enterprise Linux 3.0**
- **XML-based configuration tree**
- **Cluster monitoring tools, batch queue systems and other useful software packaged as "rolls" which can be selected during installation**
- **http://www.rocksclusters.org**

Installing a Single Cluster

- **Single cluster installation with Rocks is relatively straightforward:**

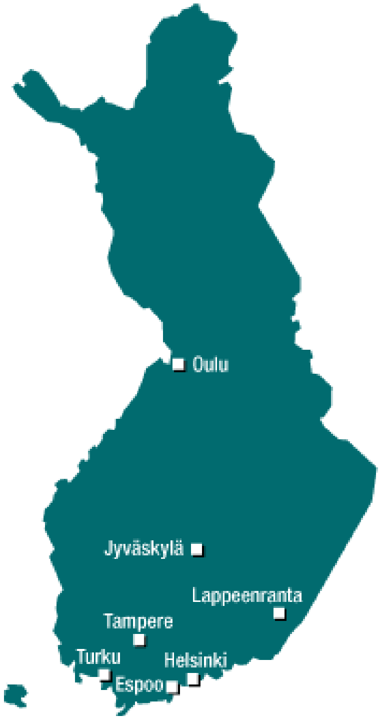
  1. Download a cd set or a smaller boot cd
  2. Install and configure the cluster front end
  3. Power up compute nodes one by one and let them install the local OS over the network from the frontend
     - frontend gives ips and stores the mac addresses automatically
  4. Test the installation, start computing

- **Well described in the Rocks manual => won't go into details in this presentation**

CSC

Finnish Material Sciences Grid (M-grid)

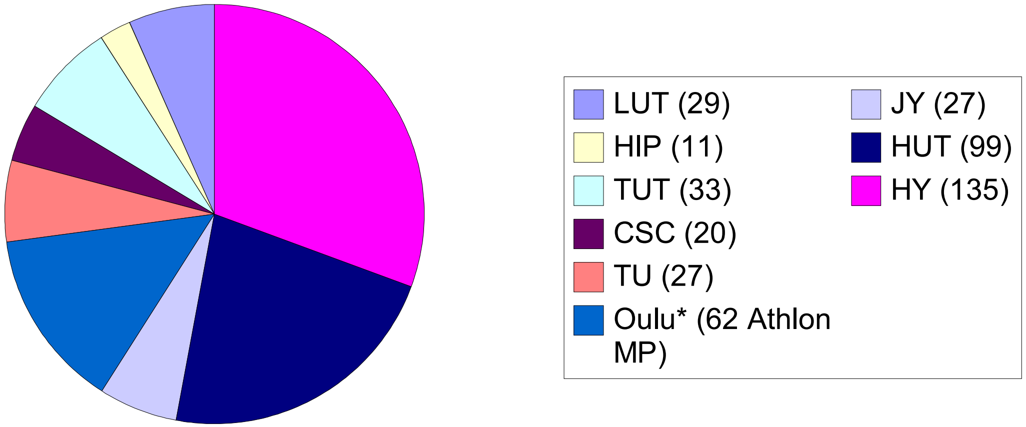
- **Joint project between seven Finnish universities, Helsinki Institute of Physics and CSC**
- **Jointly funded by the Academy of Finland and the participating universities**
- **First large initiative to put Grid middleware into production use in Finland**
- **Based on GNU/Linux clusters, targeted for throughput computing, serial and ”pleasantly parallel” applications**
- **Users mainly physicists and chemists**
- **http://www.csc.fi/proj/mgrid/**



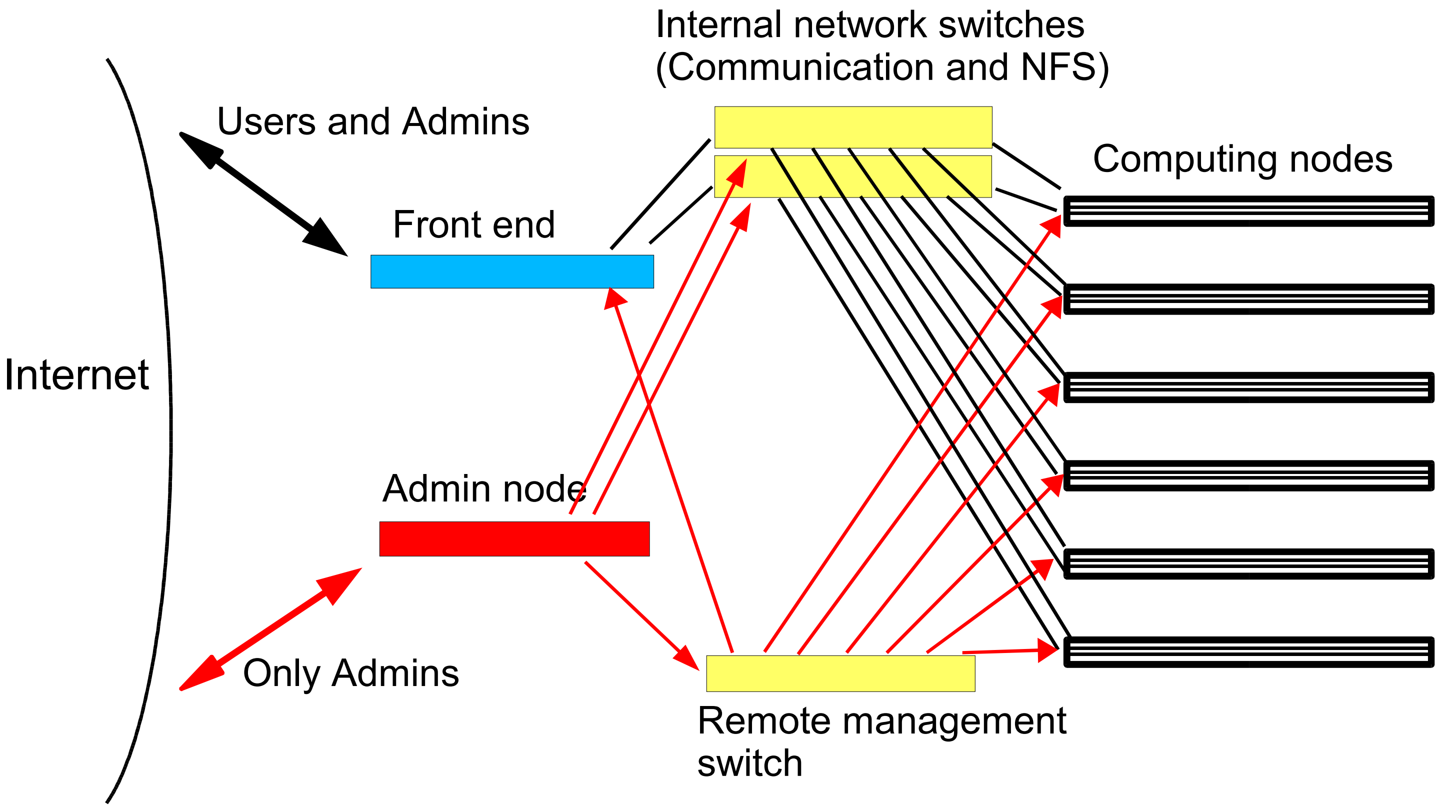
CSC

M-grid Hardware and CPU Distribution

- **Dual AMD Opteron 1.8-2.2 GHz nodes with 2-8 GB memory, 1-2 TB storage, 2xGbit Ethernet, remote administration**
- **Number of CPUs: 410 (computing nodes only), 1.5 Tflops theoretical computing power**
- **9 sites, size of sites varies greatly**



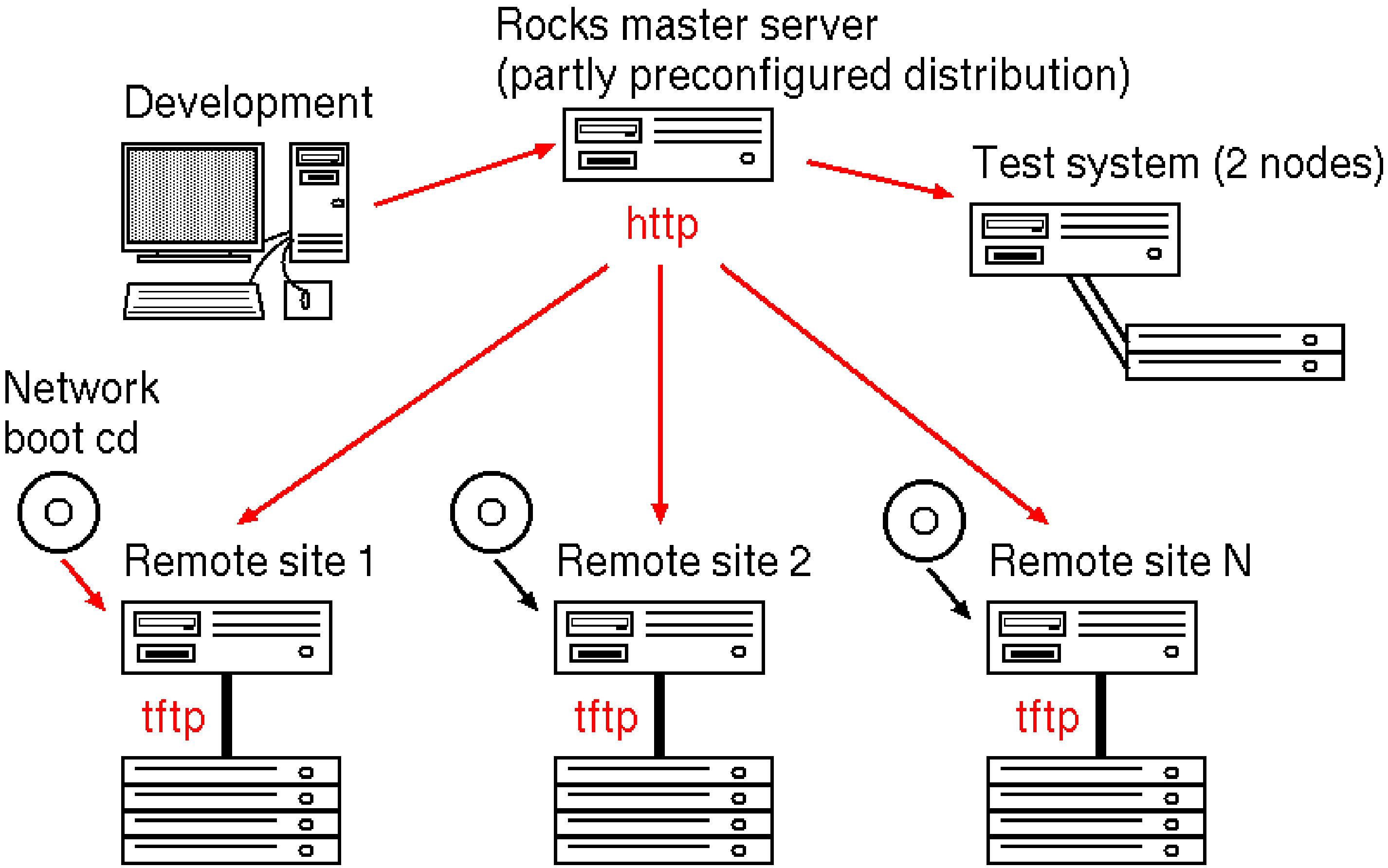
One M-grid Cluster



System Administration in M-grid

- **Tasks divided between CSC and site administrators**
- **CSC administrators**
  - Maintain (remotely) the OS, LRMS, Grid middleware, certain libraries for all sites except Oulu
  - Separate mini cluster for testing new software releases
- **Site administrators**
  - Local applications and libraries, system monitoring, user support
  - Most site admins are researchers working for the department or lab, not I.T. (but are quite competent)
- **Regular meetings of administrators, support network**

Installation Plan



Installing the Rocks Master Server

- **A standard GNU/Linux box with a web server, http and rsync open to the clients**
  - We installed it as a Rocks frontend without nodes (shutting down a few unnecessary services), but could also be some other distribution
- **Rocks distribution mirrored from the Rocks main site as the basis for development**
  - Rocks Makefile conventions and build system needed some studying to get used to
- **For development it is convenient to make a frontend boot cd which automatically sets the ip address and starts the install from the master server over the network**
  - Just append the relevant kernel parameters in isolinux.cfg: central=your_master_server_name ip=... gateway=... netmask=... dns=... frontend ksdevice=eth1 ks=http://master_kickstart_file

Customizing Rocks

- **Basic customizations in one cluster fairly easy**
  - E.g. changing compute node disk partitioning, adding software packages, adding monitoring components
  - Large parts of configuration stored in a MySQL database
- **Customizing the XML configuration tree more difficult**
  - Flexible, but needs work to get familiar with it
  - Red Hat kickstart file is generated from the XML files
- **Especially front-end installation hard to debug: a typo in the XML can make installation fail and system reboot**
  - Most people don't make customizations before installing the frontend, so there is little documentation for it
  - Developers on the mailing list have been helpful

Our Customizations

- **Main principle: Don't touch the Rocks base distribution but do all customizations as an additional "roll"**
  - Basic localization: Finnish keyboard layout etc.
    (This has been improved in Rocks 3.3.0, we started with 3.2.0)
  - Separating NFS and MPI traffic to two networks
  - Additional packages: Numerical libraries, a proprietary Fortran compiler, NorduGrid ARC middleware (soon)
  - Firewall settings, a preconfigured administrator account
- **Problem: The configuration is parsed at the installation stage when many things aren't yet in place**
  - Not nice when combined with poor debugging => ended up writing scripts which are executed at first boot after installation

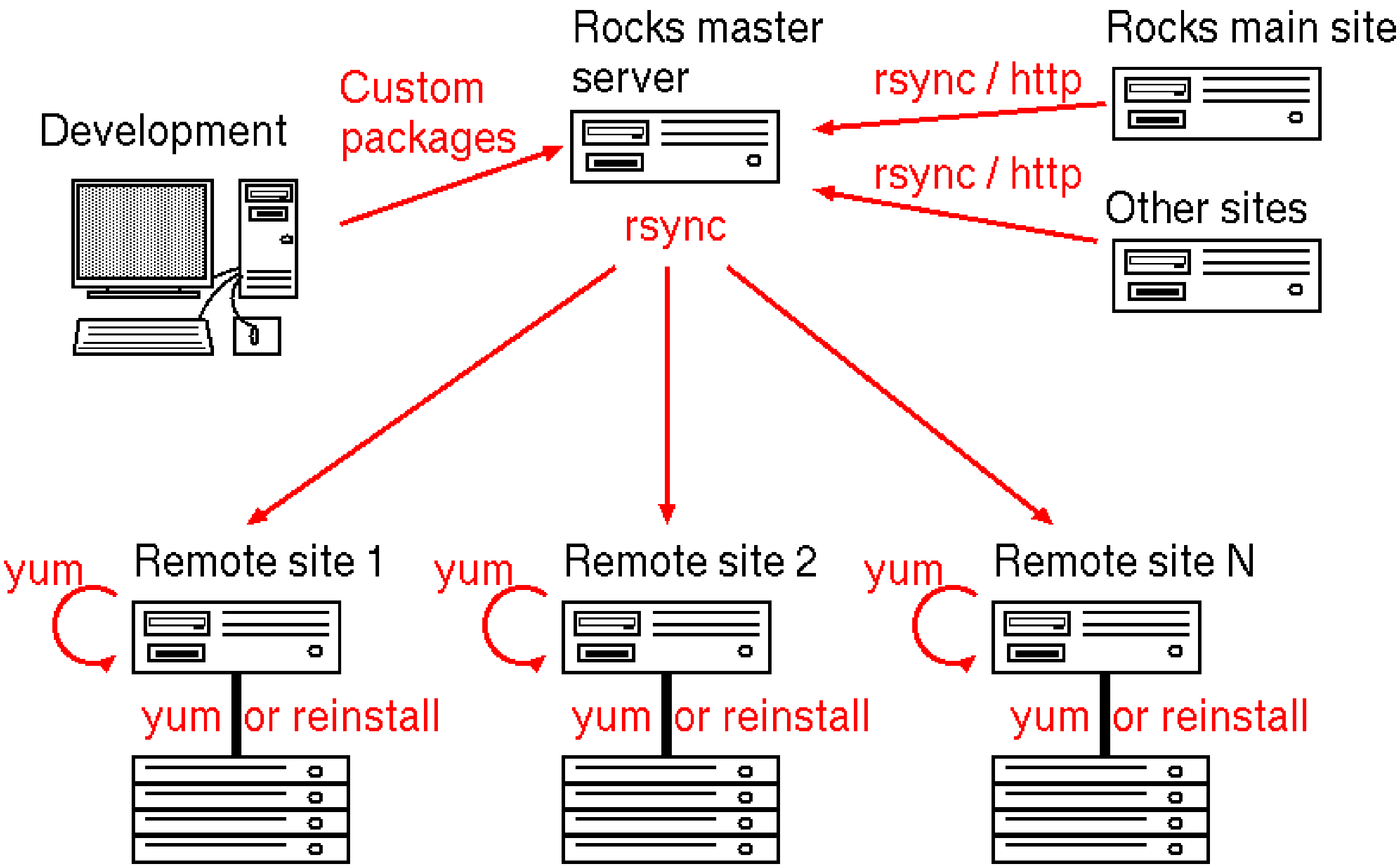
Deployment Experiences

- **CSC prepared the distribution and a boot cd, local administrators were responsible for installing their cluster**
  - One installation was done completely off-site using remote administration hardware (our front ends can boot from virtual media and we have remote access to power and console)
- **Preparing the distribution took more time than expected**
- **Actual deployment went quite smoothly**
  - Most sites spent from a few hours to one day installing the OS and nodes, larger sites took two days
  - One site had strange problems taking more time
- **Some things needed fixing afterwards as the distribution was not preconfigured as well as we planned**

Updates

- **The Rocks team doesn't currently provide security advisories or updates between releases**
  - Critical security updates need be recompiled from RHEL source RPMs or fetched from other sources
- **"The Rocks way" is to simply reinstall compute nodes whenever updates or configuration changes are needed**
  - We wanted to install some packages without reboot
- **Yum is convenient but does not support a three level structure => use rsync to mirror packages, then install using yum to both frontend and nodes**
  - Another nice way would be to schedule updates in nodes as high priority reinstall jobs which are executed when computing jobs finish (probably our next step)

Installing Updates



Rocks Pros and Cons

Good:

- **Easy to get started, designed for clusters**
- **Nice monitoring tools, many things work out of the box**
- **Many big vendors have their hardware certified for RHEL => Rocks usually works too**
- **Competent people on the mailing list**

Something to improve:

- **Security updates not provided by the Rocks team**
- **Hard to diagnose and debug installation problems when customizing the distribution**

Rocks Compared to Debian FAI

(Warning: My hands-on experience with FAI is 2-3 years ago)

- **Both Rocks and FAI (Fully Automatic Installation) for Debian GNU/Linux are based on installers, not imaging tools**
- **Rocks is a bit easier to get started with: tftp, nis, ganglia monitoring etc. are preconfigured**
- **Debian FAI installation structure is easier to understand, customize and debug**
  - FAI can be easily bent to exotic things by replacing parts of the standard installation sequence with custom scripts
- **FAI is a natural choice for Debian fans, OSCAR is another popular Free package in the "rpm world"**

More Information

- **Rocks home page: http://www.rocksclusters.org**
- **M-grid home page: http://www.csc.fi/proj/mgrid/**
- **These slides: http://staff.csc.fi/ajt/presentations/Deploying_M-grid_with_Rocks_2005-02-26.pdf**
- **Technical contact people at CSC:**
  - Arto Teräs <arto.teras@csc.fi>
  - Olli-Pekka Lehto <olli-pekka.lehto@csc.fi>
- **Thank you! Questions?**