Integrated cluster management at Manchester

Andrew McNab, Alessandra Forti
School of Physics and Astronomy, University of Manchester, Oxford Road, Manchester, UK, M13 9PL.

Email: andrew.mcnab@manchester.ac.uk

Abstract. We describe an integrated management system using third-party, open source components used in operating a large Tier-2 site for particle physics. This system tracks individual assets and records their attributes such as MAC and IP addresses; derives DNS and DHCP configurations from this database; creates each host’s installation and re-configuration scripts; monitors the services on each host according to the records of what should be running; and cross references tickets with asset records and per-asset monitoring pages. In addition, scripts which detect problems and automatically remove hosts record these new states in the database which are available to operators immediately through the same interface as tickets and monitoring.

1. Introduction
The huge data volumes produced by the particle physics experiments at the LHC at CERN have required a corresponding increase in the computing resources needed by particle physics researchers around the world, and led to the creation of grid collaborations, such as GridPP in the UK[1]. The University of Manchester particle physics group has operated the Manchester Tier-2 as part of GridPP since 2005, initially with 2000 processors on 1000 hosts. As core densities have increased, we have been able to reduce the number of distinct hosts to 179 whilst increasing the number of cores to 1624.

Computing at this scale requires automated tools to install, configure and monitor hosts and services, and we have adapted a mixture of operating system components and open source tools to address this requirement. Central to the system is a database of assets, such as hosts, switches, and racks, and one of our design principles is that the definitive copy of each piece of information (for example, an IP address) is stored in one place in this database and that all other copies are derived from it automatically. A second design principle is that monitoring and configuration should be integrated, as the resources to be monitored on a host depend on its particular configuration, and equally that faults identified by monitoring may require a temporary change in configuration such as putting a host offline.

2. Record keeping and network addresses
The database of hosts and other assets is maintained using Asset Tracker (AT), a component of the Request Tracker (RT) issue tracking system[2]. Standard records for each host include its name, ethernet MAC, IP address. We have added fields for the operating system version, hardware type, role, PBS job execution state and reason (for example, offline due to disk errors), and reinstallation status. The RT system allows the creation of tickets when faults are observed or when desired features are requested, and stores updates from the same or a different operator submitted either via its web interface or by email. As AT is part of RT, we can readily link these tickets with particular hosts, for easy cross referencing. We also link hosts with switches they depend on and physical racks they are a
member of, and this is presented in RT’s web interface for operators as a hyperlink to facilitate cross referencing. Additionally we add links from each host to a corresponding monitoring web page, which are presented in both systems’ web interfaces as direct hyperlinks.

The first two columns of Figure 1 shows how the AT database is connected to the services which are used to manage the site. As we have a full database of all host names, MAC and IP addresses, we are able to automatically generate the configuration files the DHCP and DNS services which hosts use to acquire and query network addresses. In practice this means that adding or renaming a host just requires updating its records and prompting a rebuild of these configuration files. The AT system uses MySQL[3] as its underlying database, and the scripts running on the servers dedicated to DHCP and DNS are allowed read-only access to the relevant database tables over the network.

![Diagram of system components](image)

**Figure 1: Components of the system**

### 3. Installation and initial configuration

The rightmost column of Figure 1 shows the components on a Worker Node where jobs execute and how they are connected to the site-wide services. A similar set of components exist on the grid middleware and storage hosts. When constructing the DHCP configuration files, the correct file paths and names required by the PXE boot ROMs on these hosts are included, taking into account the operating system version recorded in AT for that host. These files are fetched via TFTP. A wrapper script around the TFTP service refuses to serve the installation files if the Reinstall flag is not set in AT for that host, and the host then defaults to booting from the existing installation on its local disks.

If installation is allowed, the RedHat pxelinux boot loader is run which in turn loads the Scientific Linux[4] version of the RedHat Anaconda installation kernel. This then requests a kickstart file which...
gives desired network and disk settings, and lists packages to be installed. The kickstart file is obtained via HTTP, but the URL given is in fact a CGI script which queries the AT and returns the correct kickstart file for that host's operating system. This script also cancels the Reinstall flag to prevent an endless cycle of reinstallation. The RPMs files are obtained directly from a conventional web server as static files which allows large numbers of hosts to be reinstalled in parallel.

Once the host is installed, additional configuration if provided by cfengine with scripts and configuration file templates from a central CVS repository. Groups and classes of hosts are defined to match those assigned to individual hosts in the AT database. cfengine continues to update the configuration of the host as necessary over time, in response to changes in the central repository.

4. Monitoring and automated reconfiguration

We use Nagios[5] to monitor the state of hosts and services, both via its web interface and with email alerts which it sends after detecting critical problems. We enable both remote tests, such as whether a host responds to ssh connections, and local tests using the Nagios NRPE service running on each host. In addition to standard local tests provided with Nagios, we have written custom checks applicable to our system.

In particular, the script “man-SanityChecks” carries out locally defined tests such as the status of the /dev/null device file which has been overwritten occasionally in the past. This script is able to force a host offline in the PBS/Torque job submission system[6] if critical problems are detected, and this allows us to prevent machines from persisting as “black holes” which attract and then immediately kill all jobs which enter the cluster.

The configuration of the Nagios system itself is derived directly from the records in the AT database, again with a script which interrogates the database over the local network to construct the list of hosts, grouped by their roles and physical rack. These groupings are used by Nagios to determine which monitoring tests to carry out, and to present the hosts logically in the Nagios web interface. As with the AT interface, we include hyperlinks from each host's monitoring page to its AT web page, which allows us to check the intended role and state of a problem host and to examine the tickets which have been associated with it.

5. Experiences and conclusion

Our main observation in operating the Tier-2 facility has been that accurate monitoring is a key component in keeping production services online with good efficiencies. Accuracy requires both the speedy detection of relevant faults, but also the avoidance of false alarms and long standing “known problems” which are visible in the monitoring interface. These last two make it harder for human operators to notice genuinely new problems when they arise. For this reason, integrating the configuration and monitoring system to achieve accuracy by representing the details of different hosts’ status has been an important contribution.

References
[1] D. Britton et al, 2009, GridPP: the UK grid for particle physics, *Phil. Trans. R. Soc. A* 367, 2447-2457
[2] J. Vincent et al, 2005, *RT Essentials* (Sebastopol, CA: O'Reilly)
[3] R. Dyer, 2008, *MySQL* (Sebastopol, CA: O'Reilly)
[4] Scientific Linux collaboration, www.scientificlinux.org
[5] T. Dondich, 2006, *Networking Monitoring with Nagios* (Sebastopol, CA: O'Reilly)
[6] G. Staples, 2006, in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing* (New York, NY: ACM)

Acknowledgements
This work has been supported by the Science and Technologies Facilities Council, the University of Manchester, and the EGI-InSPIRE project.