RACF's PXE Installation Management System

Christopher Hollowell <hollowec@bnl.gov>
RHIC/US-ATLAS Tier One Computing Facility
Brookhaven National Laboratory

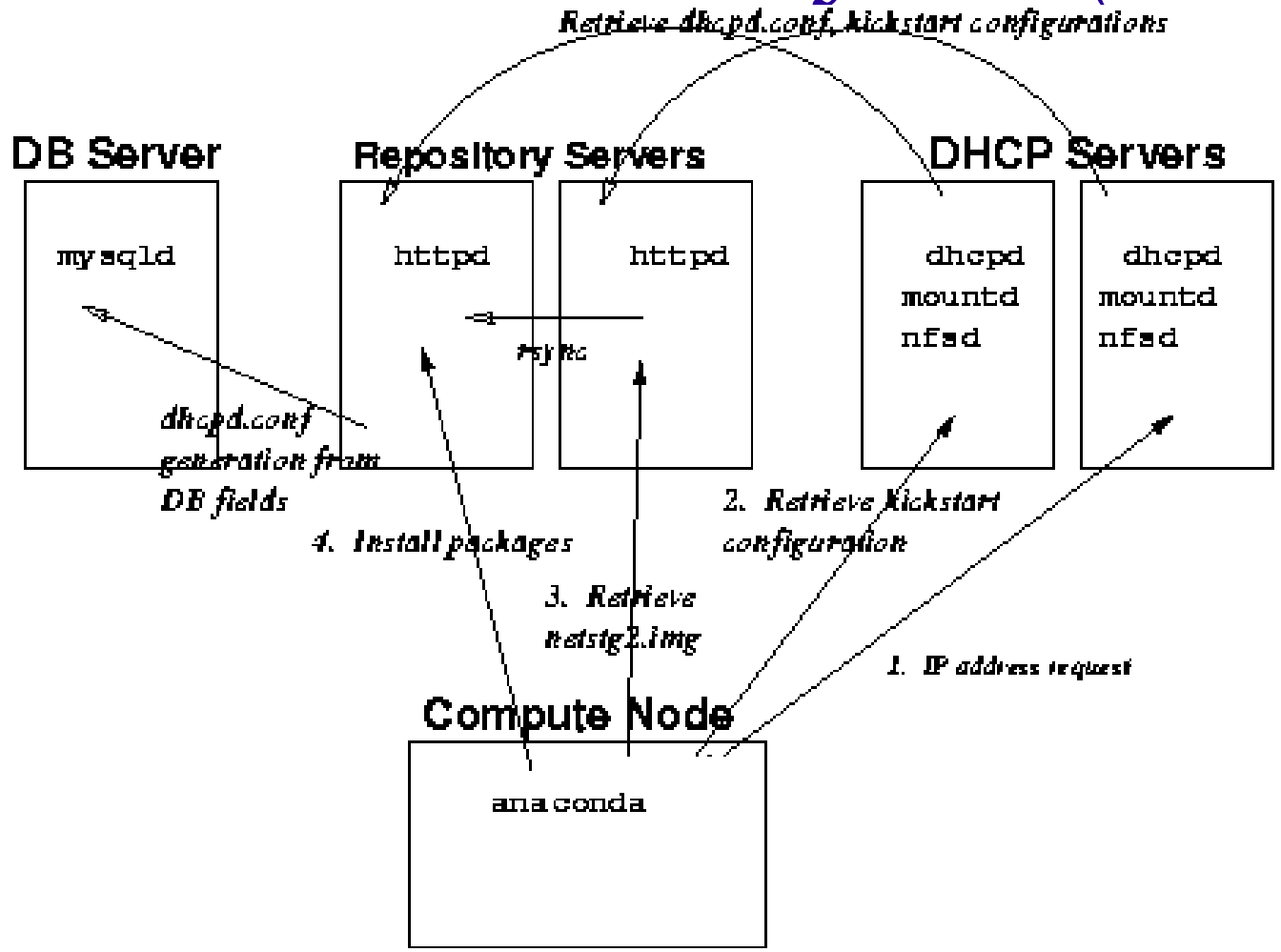
RACF Background

- RHIC/US-ATLAS Tier One Computing Facility (RACF)
- Created in the mid 1990's to support the RHIC experiments at Brookhaven National Laboratory
- Expanded in the late 1990's: became the tier one computing center for ATLAS in the US
- ~4 PB of data stored in tape silos
- Central Analysis/Reconstruction Server (CAS/CRS) farm consists of ~2000 multiprocessor/multicore hosts for data processing
  - ~400 TB of data cached in the local drives of the processor farm: managed by dCache, XROOTD, ROOTD

CAS/CRS Farm Legacy Automated Network Installation System

- Components:
  - Based on Red Hat Kickstart
  - Backend MySQL inventory database providing specific information on node configuration: hostname, NIC MAC address, IP address, netmask, physical location, hardware setup, etc.
    - Repository servers for package distribution via HTTP
    - DHCP/NFS servers providing fixed-address IP assignment and access to Kickstart configuration files
      - dhcpd.conf contains an entry for each host in the inventory database: automatically generated from DB fields

CAS/CRS Farm Legacy Automated Network Installation System (Cont.)



Legacy Installation System Architecture (fig. 1)

CAS/CRS Farm Legacy Automated Network Installation System (Cont.)

- Initially developed in 2001 with <500 systems in production
- Problem: requires boot media (single floppy/cdrom/USB key) to initiate the network install process. Not scalable
- Solution: migrate the existing infrastructure to a PXE-based installation system (completed Fall 2005)

What is PXE?

- PXE – Preboot eXecution Environment
  - Environment for booting a system via network card firmware
- Booting a server via PXE (conventional DHCP/TFTP config)
  - NIC PXE option-ROM is executed during system bootup
  - Firmware sends a DHCP request to obtain IP address, other network settings, and path to the Network Boot Program (NBP)
  - Loads NBP into RAM via TFTP and executes
- NIC doesn't support PXE?
  - Media bootable PXE stacks available:
    - Etherboot
      - http://www.etherboot.org
      - http://rom-o-matic.net

What is PXE? (Cont.)

- What can be used as the NBP?
  - PXELINUX
    - Effectively a PXE-bootable version of SYSLINUX
    - Freely available
      - http://syslinux.zytor.com/pxe.php

Requirements

- Data locally cached on the farm is mapped to particular hostnames/IP's: non-fixed-address IP assignment is not possible
- Various special purpose machines exist on the same subnets as the processor farm: do not want any machine which attempts to boot via PXE to re-install itself
- Want to be able to force re-installation of a node without physical intervention, independent of software state (i.e. even if the local OS will not completely load): requires that the BIOS attempt to boot via PXE before HDD.
  - Out of band system management software (racadm, xCAT, ipmitool, etc.) allows for remote invocation of hard resets when necessary

Requirements (Cont.)

- Large amount of local data on the farm which normally needs to be preserved during upgrades/re-installations. This requires tight management of the installation process. Physical media insertion = tight management. Can be achieved with software, but would need to be written or obtained from another institution
- Systems should not re-install themselves during each boot
  - Avoids lengthy startup times
  - Keeps temporary changes to hosts from being undone
  - Ensures data integrity
    - Data preservation managed by flags in the Kickstart configuration files

Requirements (Cont.)

- Need to be able to be able to specify machines to install, and have easy access to the list of hosts slated for installation

Alternatives

- Why not use one of the mid/large-scale cluster/installation management systems which are publicly available?
  - Additional flexibility
  - Able to reuse most of our legacy system's components
  - Already have a cluster management solution we are comfortable with: need installation management only
    - Scyld ClusterWorks HPC, SDSC ROCKS
      - Our processor farm setup does not conform to a typical single-head node ”beowulf-style” architecture
      - Scyld
        - Not currently available free of charge
        - Scalability issues
        - No need for a distributed unified process space

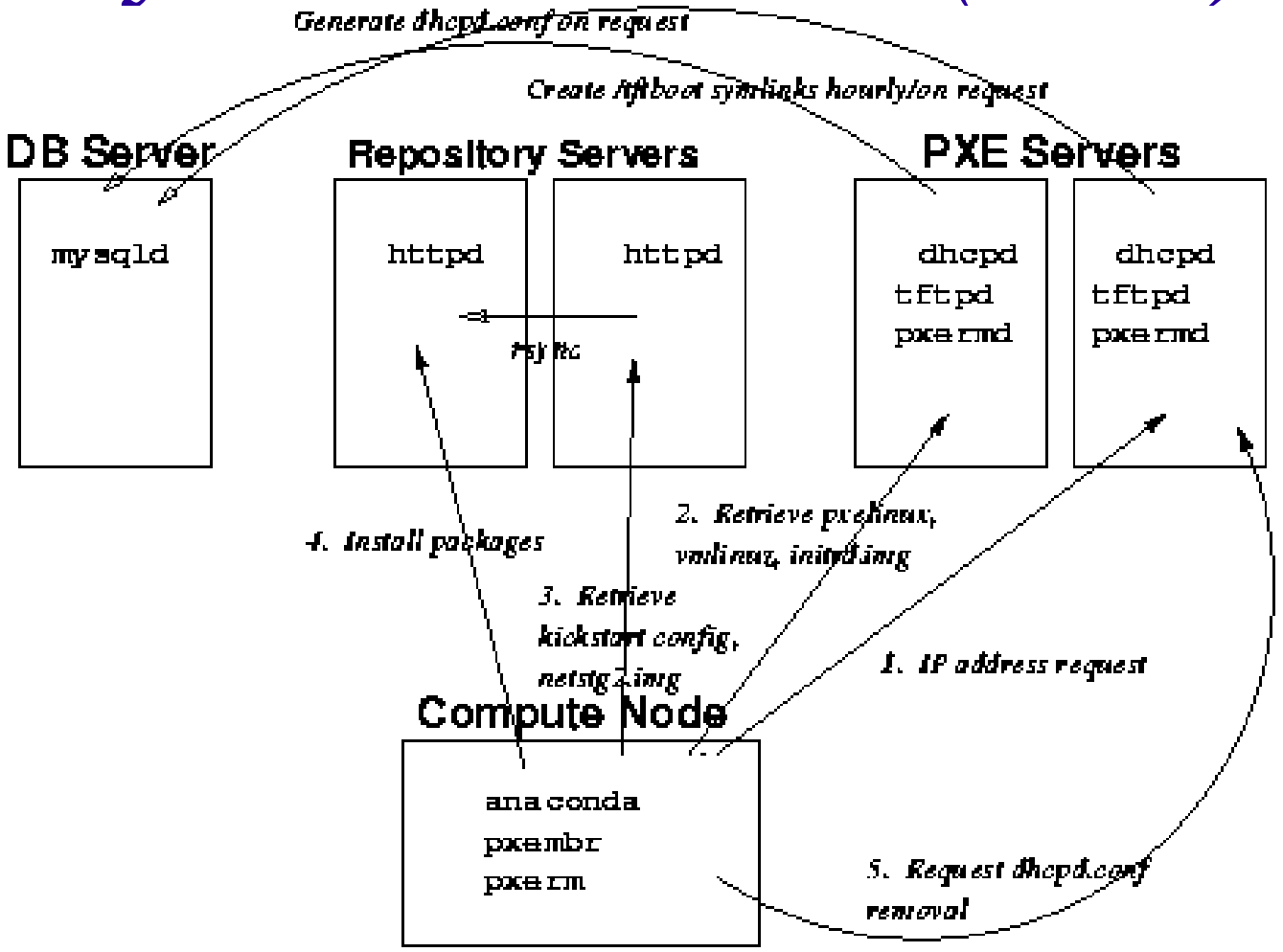
Alternatives (Cont.)

- ROCKS
  - Want our systems to attempt to boot via PXE before HDD: ROCKS will repeatedly re-install
- Quattor
  - Extremely powerful, scalable; however, with this power comes complexity
    - User guide + PAN language specification > 115 pages
    - Don't require functionality besides AII

CAS/CRS Farm Automated Installation System Architecture

- Components:
  - Several reused from the legacy system
    - Red Hat Kickstart
    - MySQL inventory database
    - Repository servers
  - PXE servers: DHCP servers now run TFTPD to distribute PXELINUX
    - PXEUtils installed on the DHCP servers: software developed to manage PXE installations
    - Use of Cisco's DHCP ”helper-address” functionality to reduce the required number of servers

CAS/CRS Farm Automated Installation System Architecture (Cont.)



Installation System Architecture (fig. 2)

PXE Server Configuration

- /tftpboot (TFTPD data area) on the PXE servers contains boot kernel, initrd.img, pxelinux.0 (NBP)

```
pxeserv# ls /tftpboot
initrd.img    pxelinux.0   pxelinux.cfg   vmlinuz
```

- /tftpboot/pxelinux.cfg directory contains PXELINUX configuration files and a symlink to the appropriate file for each MAC address in the inventory database: created by PXEUtils

```
pxeserv# ls -l /tftpboot/pxelinux.cfg
lrwxrwxrwx  1 root root  24 Jul 1 2005  01-00-a1-b2-c3-d4-23 -> pxelinux-ks1.cfg
lrwxrwxrwx  1 root root  24 Jul 1 2005  01-00-bb-aa-32-d4-30 -> pxelinux-ks2.cfg
...
-rw-r—r--   1 root root  24 Jul 1 2005  pxelinux-ks1.cfg
-rw-r—r--   1 root root  24 Jul 1 2005  pxelinux-ks2.cfg
...
```

PXE Server Configuration (Cont.)

- Several different Kickstart configurations used by subgroups of the CAS/CRS farm (distinct package sets, partition layouts, etc.)
    - One PXELINUX configuration file in place under /tftpboot/pxelinux.cfg for each Kickstart configuration

/tftpboot/pxelinux.cfg/pxelinux-ks1.cfg:

```
default ks
prompt 1
timeout 300
display boot.msg
label ks
    kernel vmlinuz
    append ks=http://repo.example.com/pxe/ks1.cfg ksdevice=eth0 initrd=initrd.img
```

/tftpboot/pxelinux.cfg/pxelinux-ks2.cfg:

```
default ks
...
label ks
     kernel vmlinuz
     append ks=http://repo.example.com/pxe/ks2.cfg ksdevice=eth0 initrd=initrd.img
```

PXE Server Configuration (Cont.)

- /etc/dhcpd.conf
  - Specifies the path to pxelinux.0
  - Only contains entries for hosts which are to rebuilt
    - Entries added/removed by PXEUtils

skeletal /etc/dhcpd.conf:

```
default-lease-time 0;
max-lease-time 7200;
option domain-name-servers 10.0.0.1 10.0.0.2 10.0.0.3;
option domain-name ”example.com”;
ddns-update-style none;
filename ”pxelinux.0”;

subnet 10.1.0.0 netmask 255.255.254.0 {
     option routers 10.1.0.4;
     option subnet-mask 255.255.254.0;
}
```

PXEUtils

- Set of six programs for managing PXE installations
  - Written in C and Python
  - Compact: ~1500 lines of code
- Tools
  - pxestat
    - Allows one to start, stop, or check the status of the required daemons on a PXE server
  - pxectl
    - Used to add/remove hosts to/from dhcpd.conf on a PXE server and restart DHCPD

PXEUtils (Cont.)

- pxermd
  - Daemon which listens on a PXE server for hosts' requests to be removed from dhcpd.conf after their Kickstart installation has completed. This process spawns a child which checks for changes to dhcpd.conf every 5 minutes. If a change is detected, it restarts DHCPD
- pxerm
  - Client software run during the Kickstart postinstall stage which contacts pxermd to report installation completion

PXEUtils (Cont.)

- mkpxelinks.py
  - Python Script which scans a specified database and creates symbolic links to PXELINUX configuration files under /tftboot/pxelinux.cfg for each host/MAC address it encounters. Run hourly by cron; force execution by passing '-l' to pxectl
- pxermbr
  - Invalidates the MBR on a specified drive such that the BIOS will not boot off of it. It is necessary to run this program on hosts where PXE is listed after HDD in the BIOS boot sequence (few systems in our environment)

The Installation Process

- Check if the required daemons are running on the PXE server:

```
pxeserv# pxestat
dhcpd: stopped
xinetd: stopped
pxermd: stopped

PXE INSTALLATIONS DISABLED
```

- If they're not, start them

```
pxeserv# pxestat on
Starting dhcpd:                    [ OK ]
Starting xinetd:                   [ OK ]
Starting pxermd:                   [ OK ]

SUCCESS
```

- Ensure the /etc/tftpboot/pxelinux.cfg symlinks are synchronized with inventory DB information:

```
pxeserv# pxectl -l
```

The Installation Process (Cont.)

- Specify machines to install

```
pxeserv# pxectl -r -a node0010 node0100-0200
node0010
node0100-0200
Shutting down dhcpd:                [ OK ]
Starting dhcpd:                     [ OK ]
```

- The machines are added to the DHCPD configuration file via information obtained from the inventory DB

/etc/dhcpd.conf:

```
default-lease-time 0;
max-lease-time 7200;
option domain-name-servers 10.0.0.1 10.0.0.2 10.0.0.3;
option domain-name ”example.com”;
ddns-update-style none;
filename ”pxelinux.0”;

subnet 10.1.0.0 netmask 255.255.254.0 {
     option routers 10.1.0.4;
     option subnet-mask 255.255.254.0;
          host node0010 {
               hardware ethernet 00:11:22:33:44:55;
               fixed-address  10.1.0.20;
          }
          host node0100 {
               hardware ethernet 00:22:33:44:55:66;
               fixed-address 10.1.0.110;
          }
     ...
```

The Installation Process (Cont.)

- If a system doesn't support booting via PXE before HDD, invalidate the host's MBR

```
node0010# pxembr -p /dev/[h|s]da
```

  - Since we want to preserve the partition table in the MBR, this command invalidates the boot record by changing its magic number (from 0xAA55 to 0xAA50)
- Boot/reboot the systems slated for installation
- PXE firmware executes, retrieves pxelinux.0 via TFTP
- pxelinux.0 is executed and downloads the configuration file named (via symlink) with the MAC address of the host it is executing on (dashes between the octets, and starting with 01-) from the /tftpboot/pxelinux.cfg directory on the PXE server

The Installation Process (Cont.)

- PXELINUX retrieves the specified boot kernel and initial ramdisk from the PXE server and executes the kernel
- The initial ramdisk contains the Anaconda startup code which retrieves the Kickstart configuration file, and netstg2.img
- Anaconda executes the %pre scripts in the Kickstart configuration file
  - Runs pxembr with '-m' option to restore the MBR magic number if necessary
  - Examines /proc/partitions and generates Kickstart drive partitioning directives, based on a data preservation flag. These directives are incorporated into the Kickstart configuration file via a %include statement

The Installation Process (Cont.)

- Requested packages are installed
- %post scripts execute
  - Calls pxerm: opens a connection to pxermd on the PXE server, indicating installation has completed
- pxermd removes the host from dhcpd.conf, restarts DHCPD
- Node reboots
  - Times out when it attempts to boot via PXE
  - Boots host OS off the local disk