Disk Server Deployment at RAL

Castor F2F

RAL - Feb 2009

Martin Bly

Science & Technology Facilities Council

GridPP
UK Computing for Particle Physics

Overview

- General Deployment
- Castor Specifics
- Issues

General Deployments I

- PXE/Kickstart
- Two variants
  - Version 1 three stages:
    - Kickstart (reboot)
    - Updates
    - Personality (reboot)
  - Version 2 (scriptlets):
    - Kickstart (reboot)
    - Updates + personality (reboot)
- Version 1 used for disk servers of all types
- Kickstart, updates, personalities, scriptlets are all hand crafted
- Update script is ‘standard’ for all OS types and variants
  - Add repository definitions, brings OS up-to-date and installs standard requirements (AFS, SSH keys…)
  - Also removes some unwanted stuff (Bluetooth, Samba, WiFi…)

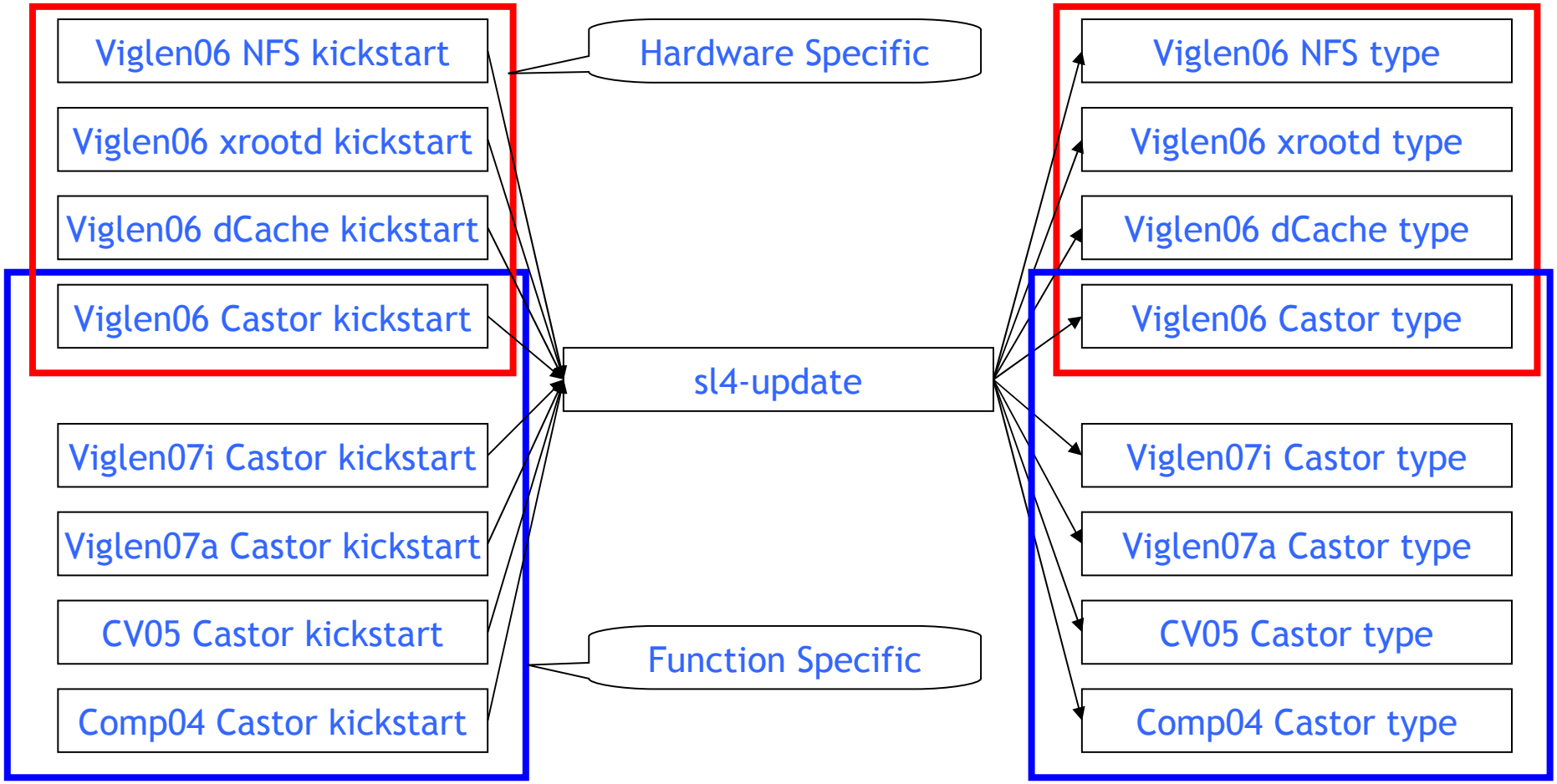
General Deployments II

- Personality scripts
  - Make the systems into BDII or WN or CE or disk server or a specific type
  - Additional repositories (Castor, Glite, …)
  - Installs personality specific packages
  - Removes additional OS stuff some variants of same type might need but are unnecessary in a specific cases
  - Configures system
    - Adds or edits config files

General Deployment III

- Kickstart files are hardware specific
- Kickstart files are task (personality) specific
- All OS variants call the same update script
- Then call the task-specific personality script for the specific hardware
- System has grown to the way it is over time
  - Now: less than ideal

General Deployment IV



Castor Specifics I

- Castor disk servers and central services hosts augment the standard deployment using Puppet to configure stagemap.conf etc.
- Disk servers deployed in two stages
- Detailed procedure to follow for complete process – lots of checking
- Stage 1: Install to 'nonProd' class (per VO)
  - nonProd – holding class for servers ready to go into production
    - 'nearline' – but not quite as near as I'd like.
  - Careful choreography of changes to Puppet, LSF
  - Uses the Kickstart system for main provisioning
    - Master config file – Castor version, VO, Service Class – use dby kickstart for VO and service class specific actions
  - Certificates added by hand
  - Registered with rmMaster
  - Pause to check (test!) all is OK.
  - At this point, server is being monitored by nagios, ganglia etc
    - In case it needs an intervention

Castor Specifics II

- Stage 2: Deploy to production service class
  - Configure Puppet to declare Service Class
  - Configure Puppet to add server to LSF production configuration
  - Enable new server on Castor central server
  - Activate Nagios callouts
  - Admire the data flow… ☺
    - (Cacti, ganglia etc)

Issues

- Still a significant number of manual steps on LSF system and central services to add a new server
  - Particularly configuring Puppet manifests
  - Choreography of steps on diverse systems otherwise ‘the logs fill up’!
- Status
  - Replacing spreadsheet with a database
- We don’t have an ‘simple’ end-to-end system that can orchestrate the changes needed to (re)deploy a server
  - Do we have the optimum MO for Castor?
    - Will it scale?
  - Working on our Fabric Management options
    - Need to understand if the sequence of steps for deployment is the best possible (if the MO is correct)?