Enabling Grids for E-sciencE

Tools and techniques for managing virtual machine images

Andreas Unterkircher, Dimitar Shiyachki
CERN Grid Deployment Group
Havard Bjerke

www.eu-egee.org

Content

- **Motivation**
- **Tools developed**
  - VM image generation (libfsimage)
  - GUI for VM image generation (OSFarm)
  - Efficient transfer of VM images

gLite certification process

- **GLite uses a continuous release process. Services are updated individually on top of a baseline release.**
- **Updates are added via a patch**
  - We use Savannah for bug and patch tracking
  - Has one or more bugs attached
  - Can also be used to introduce new features or services
  - Has all relevant information: OS, architecture, affected services, baseline release, configuration changes, rpm lists etc.
- **Patch is being certified**
  - Update and configure affected services
  - Run basic and regression tests
  - Verify if attached bugs are fixed and write regression tests.
  - Put patch into “certified” or “rejected”. For the former the patch is ready for release to pre production and later for production.

Problems in patch certification

- **Certification of several patches at the same time can cause conflicts.**
- **A non functional patch may spoil the whole testbed**
- **Patch certification often fails already at an early stage (rpm installation, configuration)**
- **A failed patch can pollute a machine. A complete reinstallation is necessary.**
- **Many scenarios and interactions have to be considered: gLite 3.1/gLite 3.2, SL4/SL5/Debian, x86/x86_64**
- **165 patches treated in 2008**

Virtualization for patch certification

- Patch certification can only be done efficiently by heavily using virtualization
- Already in summer 2006 we ran our own VM management system now known as VNode
- We collaborate with various groups in CERN IT
  - Openlab: OSFarm
  - FIO: Quattor profiles, SLC Xen support
  - Netops: hostnames for VMs

vNode

Certification process

- We operate a certification grid testbed offering all services.
- Certifiers bring up virtual machines containing the updated versions of nodes affected by a patch and connect them to the testbed.

Requirements for VM images

- We want to reproduce what is deployed at a site
  - Base OS installation + gLite node type
- Creating a VM images by scratching a physical machine is not practical
- A VM creation engine should run on an existing OS in a chroot environment
- VMs with different Linux flavors and architectures: SL(C)*, Debian,… x86 and x86_64

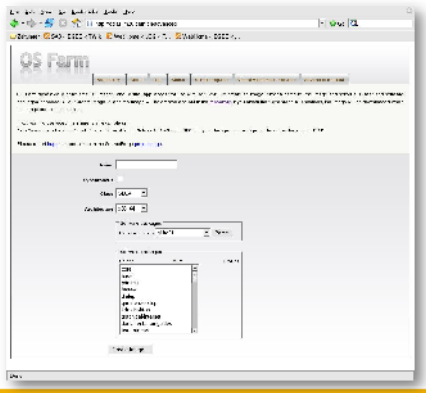
VM image generation - libsfimage

- Python library for generating Linux file systems and populating them
  - Has command line interface
  - Produces images in a chroot environment
- Produces a tar.gz file that can be used as an image to boot with Xen
- Supported distributions: SL(C)3/4/5, Debian, Ubuntu, CentOS, Fedora on x86 and x86_64
- Available in the xenvirt module in CERN's CVS

Example

```
./genfs.py
  -t SL4_x86_64
  -d /tmp/SL4fs
  -o /tmp/SL4_x86_64_img1.tar.gz
  -g Base Core
  -p python openldap-client
  -w rootpwd
```


OSFarm



- Uses libsfimage for creating core images
- GUI for configuring and adding software to images
- Optimizes the image creation process by caching shared data

OSFarm

Images and their configuration (XML) are stored in the repository for later retrieval

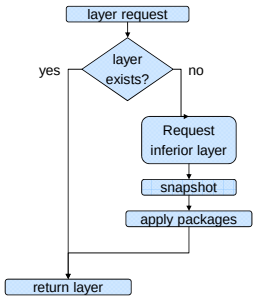
Each request is checksummed and compared to existing configurations

Layered image generation

- **Image core: minimal VM image**
  - can be shared between VM image configurations
- **Image base: contains software critical for virtual appliances**
  - Can be shared between virtual appliances
- **Image generation process is optimized by caching and sharing lower stages of an image**

Copy-on-write staging

- Base stages are kept in stage
- Uses LVM snapshots (copy-on-write) for instantaneous staging
- Tag of a cached image is the checksum of its configuration parameters

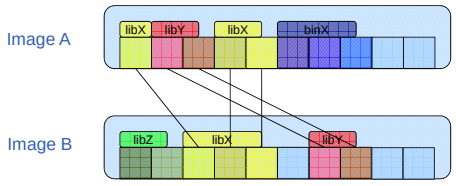


Using images on the Grid

- Many different user communities use the Grid. It is difficult for the Worker Nodes to fulfill all the different requirements.
- Provide each job on the Grid a virtual machine with a dedicated OS setup
- With thousands of users image transfer becomes an issue
- Observation
  - Images are often similar
- Content-Based transfer: don't transfer the whole image, just transfer the difference to what is already available

Content-Based transfer

- Each file starts on a block boundary
- Identical blocks can be identified with a hash checksum



Only 50% of image A needs to be transferred if image B is already at the destination

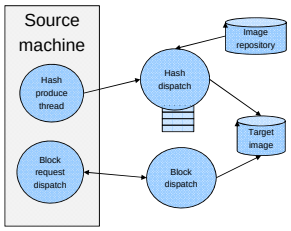
Image comparisons

- High commonality can be expected in practice
  - Different Update versions of same base OS
  - Similar applications needed by users from same research area
- SL 3 (343 MB) and SL 4 (762 MB)
  - Transfer SLC4 to SLC3
    - 22 % common blocks
  - Transfer SLC3 to SLC4
    - 48 % common blocks

Cost

- **Generating hash tables for source file and target repository – linear cost**
- **Accessing hash tables**
- **Hash table data overhead**
  - Depends on
    - Hash function (e.g. SHA is 20 bytes)
    - Block size
  - Measured 0.5 – 2.0 % of the image size

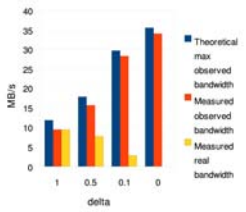
Implementation



- Prototype implementation done By H. Bjerke at openlab
- Multi-threaded
- Implemented in Java

Measurements

- Disk read speed: 35.6 MB/s
- Network bandwidth: 11.9 MB/s
- Delta = difference between source and target



Discussion