Mixing SL5 and SL6 with ‘chroot’

Brian Bockelman
GDB January 2011

Subtitle:
Running CMSSW on Fedora 16

Work done as part of the OSG Technology Area.

Motivation

- USCMS has generally enjoyed opportunistic access to other clusters at its T2 sites.
- Recently, the "other clusters" are based on SL6/CentOS6/RHEL6.
    - If CMS can run on SL6 clusters, it can access an additional 5k cores at Nebraska.
    - Purdue, Vanderbilt are also installing SL6-only clusters during Q1-2012.

Motivation

- SL6, of course, contains the new version of $THAT_SOFTWARE you really want.
- cgroups provide wonderful mechanism for controlling resources used by a process.
- Condor, as of 7.7.0, will create a cgroup per job
  - Provides accurate accounting for CPU and RAM usage, plus ability to reliably kill all processes spawned by a job.
- Since OSG switched to RPMs, no more atomic upgrades or ability to rollback.

SL6 Prospects

- When will we see SL6 worker nodes?
  - CMS will start to validate their SL5 binaries on SL6 hosts in January.
  - OSG plans on providing worker node software in first half of 2012.
- It's possible both organizations will finish very quickly.
  - Hope for the best, plan for the worst!

Chroot

- The chroot command changes the root directory seen by a process and its children.
  - So,“/chroot/sl5/usr/bin” becomes “/usr/bin” to the process.
  - See “man chroot” for more info.
- Basic idea is to place a complete copy of the SL5 userland in the /chroot/sl5 directory, so processes incompatible with the base OS can run.

Why not VMs?

- VMs have several overheads:
  - Small percentage of CPU.
  - Small-to-Medium I/O loss.
  - Inefficient memory management between multiple VMs on the system.
  - Management headaches, especially at sites that are otherwise batch-oriented.

Why not VMs?

- For example, with batch systems / chroots:
  - You can allow jobs to increase beyond their memory limits as long as there's sufficient RAM.
  - When RAM runs out, you can have the kernel only swap out (but not kill) jobs over their limit.
    - Easier to accomplish when one kernel is controlling all of the memory.
- Given proper middleware support, VMs provide inferior resource management versus other options!
- How do you do this with a batch system? Hint: read last slide.

Building a chroot

- Basic steps:
  1. Create a clean directory.
  2. Create a new RPM database (defaults to host OS format), and install yum into the chroot / new RPM database.
  3. Dump the RPM database to a text format.
  4. Remount necessary filesystems.
  5. Perform chroot.
  6. Re-load RPM database into a RHEL5 format.
  7. Do "yum groupinstall Base"
  8. Customize the chroot to your liking.

Building a chroot

- Of course, I skip quite a few details.
- Full script available on github:
  - https://github.com/bbockelm/RHEL5-chroot
- This sample includes building the chroot, installing the base OS and grid WN, and installing CVMFS.

Deployment

- How to get the chroot on your WNs? Two basic approaches:
- RPM: Create a giant RPM to version / install the entire userland.
  - Really, this is not what RPM is designed for. However, our configuration systems really love RPMs.
- rsync: Place master copy on NFS/rsync server, have a dumb / simple cronjob.
  - Power is in the simplicity, but does not play well with configuration systems!
- Currently, we ***rsync***, but plan on switching to ***RPM***.

Deployment

- Some things need to be in the host image and bind-mounted* to the chroot:
  - Home and application NFS directories.
  - /dev, /proc, /sys, and important sockets (think NSCD).
  - passwd, group, and resolv.conf files.
  - CA certificates.
- *Bind mounts are basically mirroring (binding) one part of the directory tree at a new location:
  - `mount --bind /dev     $CHROOT/dev`

Deployment Notes

- Note that we can make /chroot/sl5 a symlink, and deploy atomic update. If the RPM upgrade "goes bad", you can rollback just by changing the symlink to the previous directory

Batch System Support

- **PBS**: Set PBS_O_ROOTDIR appropriately in the job's environment.
- **Condor**: Support coming in 7.7.5; jobs request a "named chroot", and each batch slot map the name to a chroot directory.
  - Add "+RequestedChroot=SL5" to the submit file and 'NamedChroot =?= "SL5'" to the job Requirements.
- Both batch systems will invoke the chroot syscall after forking, but before 'exec'ing the user process. Batch system lives in the base, user process lives in the chroot!

Downsides

- Chroots provide a SL5 userland, but a SL6 kernels.
    - I'd say that if your application is that sensitive the kernel, there are other issues. Are you really sure you know what your T2s run?
- Deployment is admittedly awkward. Ideas?

Conclusions

- Chroots provide a powerful mechanism for sites transitioning to SL6.
  - Basically zero overhead.
  - Can be done independently of LHC experiment or grid timelines.
  - Support from batch systems; no need to deploy a new infrastructure layer.

Teaser!

- This presentation is a subset of an upcoming colloquium:
  - "*Putting Condor in a container*: Adapting virtualization techniques to batch systems"
  - Will be on Thursday of "TEG Week" at NIKHEF.
  - We will cover how we combine cgroups, chroots, and namespaces to provide powerful node resource partitioning in Condor.
- Don't miss it!