SGE Worker node Visualization

Owen Synge
Yves Kemp
Herman Hessling

Introduction

- Virtualization provides
  - Abstraction of hardware from Job OS
  - Security improvements (Washes your OS whiter!)
    - No daemons left, Clean OS every time
      - No proxy hijacking fears.
  - Potential to run legacy OS's
    - Experiments don't change code after data taking starts.
    - How redhat intends to support EL4 in long term.
  - Better partitioning (If used with VM per job).
    - Jobs cant steel memory making your job crash

Model we follow

- Concerns about client defined OS
  - Particularly in UK and USA could have legal issues
- Model 4
  - defined in DESY Virtualization Workshop
  - Multiple predefined OS's by admin
  - User Selects OS via Queue but could be extended.
  - Queue selects OS (other models possible)

Architecture

- Should be production stable
- 4 parts to project
  - SGE
    - Configure a Queue per image type
  - SGE queue to image Integration.
    - Prolog, Epilog -> Start, Stop Virtual image
  - ImageManager.
    - Wipes the image clean for each job.
  - Modified Glite image.
    - No Batch Integration installed.
      - started by SGE starter remotely via ssh.

Simple Code, for simple deployment

- http://vmimagemanager.wiki.sourceforge.net/
- 2 New components
  - Visualization abstraction
    - vmimagemanager.py
      - 1 new configuration file
  - SGE Integration scripts
    - Vmimagesgeint
      - No configuration files (though sudo must be set up)
  - Glite worker node Images
    - Standard install,
      - but with no batch queue integration

Deployment

- Planning to test with NIKEF
  - Hoping to coordinate with Denis
- Planning deploying at DESY
  - Not 100% confident, but have willing test users
- Planning deploying at FZK*
  - After speaking with us FZK has implemented a system based upon the same principles but for a different batch queue.

Concerns

- Virtulaization costs (overhead of virtual hosts)
  - Apparent increase in network latency
    - We have not benchmarked (Hope SA3 can test)
  - Bandwidth seems unaffected
- Lack of support effort maybe critical
  - Since I am on loan to dCache from SA3 we could be a little flexible, given CERN/LCG blessing.
- All batch queues need to be integrated.
- Low latency network drivers in VM codes base.

Release time-line

- Testing new vmsgeintegration scripts
  - Removed sleep from host boot up script
    - Waited 30 seconds after OS started
      - before ssh connection copied job wrapper to VM
  - Expected to be released this week.
- LVM back ending of vmimagemanager.
  - Expect to do this as soon as we have some feedback from a deployment site.
  - See no reason to add unless used.
  - Trivial to write. (would speed node cycle time)

Possible Enhancements

- Using LVM more effectively
  - LVM snapshotting etc.
- More Batch Queues.
- Detailed job monitoring/management.
  - Adding Freezing of Jobs / backfilling etc.
- Supporting more VM systems
  - KVM and Solaris Zones
- Biggest overhead is booting OS
  - Projects like upstart show system V booting is slow.

Future

- Project is simple! (And not our day jobs).
  - Based upon work developed for testing purposes
  - Also reused for dCache development teams testing
- Features will be added if requested.
  - Expect many competitors (with this as day job)
    - FZK reimplemented
      - We explained how it was designed at GridKa School.
        - but implemented on different batch queue.
      - Investigating co-operating on a paper for CHEP.
    - The idea is so simple that can easily be done.