Virtualization

Thoughts, Ideas, and a little experience from ATLAS Perspective

Amir Farbin, Alden Stradling
University of Texas, Arlington

Overview

- ATLAS Analysis Perspective
- Advantages of Virtual Machines
- Possible Scenarios
- Realities of VM Computing
- Data access performance
- Apologies if some of this is naive... or already discussed/understood.

ATLAS Software Requirements

- I will focus on Analysis-
  - In many respects Analysis requirements are more stringent than those of simulation, reco, etc...
  - More relevant for VMs deployed by users
- Analysis activity consists of:
  - In framework (athena) analysis or ROOT analysis using libraries from the framework → requires ATLAS software kit.
  - Pure ROOT analysis

    ➡ High I/O
  - Transferring data/submitting jobs to/from GRID → requires gLite + perhaps some experiment specific services
- Simulation is also interesting since there is scarcity of CPU resources. VMs may serve as a means of tapping leveraged computing.

Required Software

- ATLAS Releases...
  - Distributed in kits (fancy tar-balls) as often as weekly.
  - Only support linux. Hoping for Mac OS X for years.
  - O(10) GB... though a lighter analysis kit of O(1) GB has been promised.
  - Strong dependencies on libraries/gcc. Building code requires some tweaking of SLC to get working... simply running is easier but not always trivial.
    - Best bet is SLC4.x... though building here is still non-trivial.
- GRID software (gLite)
  - Important part of the analysis flow:
    - Build/test analysis locally (eg on small data samples)
    - Submit job over large samples on GRID
    - Collect results
  - Big limitation: gLite only supports SLC4.x
  - Installation/configuration is typically difficult. Most use it via AFS!

Advantages of VMs

- Multi-platform support: Have exact same environment on any machine (linux, mac, windows).
  - I can run/develop ATLAS SW on my laptop.
- Bundle working and validated versions of our software
  - Minimize user system preparation time.
    - Anyone can "instantly" have a working machine.
  - Minimize cluster admin duties.
- The goal is to provide an experiment-specific VM
  - Anyone can get started with little expertise...
  - Might provide a means for Universities to setup their tier 3s.

Scenarios

- *The laptop*- often not ideal to run SLC, much prefer staying in Mac OS X, or Ubuntu, etc...
  - Build code, testing on small samples, interactive analysis of data at the late stages analysis (maybe with PROOF)
  - Finite disk: Limited datasets used for testing or output of late stages of analysis
  - Submit jobs to GRID/collect highly reduced results
  - Best features: sessions survive laptop sleep/hibernation... and working on the plane.
- *The desktop*-
  - Build code, interactive and batch processing, PROOF
  - Similar to laptop, but more disk
    - Bring few TB of data for limited processing... larger datasets will require much more CPU.
    - Sometimes limited in OS by University/Lab... or may be leveraged resource.
    - Even if possible to install SLC, may not be worth the trouble to get everything working.

Scenarios

- *The Cluster*
  - Some interactive, mostly batch/PROOF processing of large datasets
  - Resources can be leveraged (shared with non-ATLAS people).
    - Ex: Existing departmental cluster running other OS
    - Limited ability to perform custom configurations
    - Resources may be allocated as needed
      - Interesting idea: Universities have 1000's of interactive windows machines which site idle at night.
  - Even if no real limitation to configuration, admin may prefer not to manage the a cluster of SLC machines.
    - Using VMs, admin doesn't need to build additional expertise on experiment specific elements.
    - Ideally just admin just needs to deploy specific pre-configured virtual appliances

Scenarios

- *Leased Resources (Cloud Computing)*:
  - eg: Amazon Elastic Compute Cloud
  - $72 - $144 per month per core... billed to the minute of usage.
  - Based on VMs.
    - Simply pay for resources as you need them.
    - This may be a great place for simulation.
- *Portable (thumb?) Drive*:
  - Keep all of your work on a drive you take with you... plug-in and work anywhere.

Realities of VM Computing

- *CPU performance*: covered by others... doesn't seem to me that there is huge loss of efficiency here.
- *Multiple VM instances*:
  - VM implementations may be limited in cores. Eg 8-core machines need to run 4 instances of vmware.
  - Or want to run on lots of different machines.
  - Issues:
    - Need common storage
    - VM Management: start/monitor/stop VMs
    - Individual VM Configuration: eg network
    - Batch software

Data Storage on VM

- One of the first obstacle we faced when trying to do real analysis with VMs was storage.
- Storage:
  - Types of data we need to store: ATLAS/GRID SW, User code, Data (AOD/DPD)
  - Possibilities: VM disk (limited access), Host disk (network, eg NFS), NAS, xrootd
- ATLAS SW
  - needs updating on frequent basis → dynamic in size (depending on # of viable current releases)
  - Fast access minimizes build times
  - Large (10's of GBs), so ideally don't want to replicate for every VM.
- GRID SW- Small.. and due to installation/configuration issues we distribute it w/ the VM

Data Storage on VM

- User code (home area)
  - Must have access across
    - VM and host- user might want to user local editor or access when VM is no on.
    - Multi-VMs- need you home area available on all the VMs
      - Should be able to upgrade VM w/o loosing user home area
      - ➡ Suggest storage outside VM
  - Data (AOD/DPDs)... similar access as user code, but much larger.
    - clearly I don't want TBs of data inside a VM disk
- We've spent a bit of time evaluating our storage options.

Setup

- I have been using VMs for nearly 2 years now... mostly for sw development/testing on my laptop.
- Hardware: Intel Core Duo MacBook (2GB/120GB), Intel 2 Core MacBook Pro(4GB/160GB), **Intel 2x 4-core Core 2 Duo (Penryn) Mac Pro(14GB/3.5TB)**.
- VM Software: VMWare fusion ($80). VMWare player is free for Windows/Linux (can't create a VM, but can run one). We've also used Parallels... but focusing on VMWare now.
- VM: SLC4, with recompiled kernel. 30 GB disk image. I've copied this VM many times between VMs (or ran multiple instances on the same host).
- Environment: Do not start X in VM (no graphics performance hit). Simply ssh to VM and use X-client in Mac OS (or linux or Windows). Gives unified environment (Ex: copy code from VM to Mac Mail).

Test Parameters

- Want to find ideal configuration for VM to read data served from host.
- Options: NFS, HGFS (vmware invention?), xrootd
- Network: NAT (packets go through emulated hardware only) or Bridged (packets got to router and back)
- Compare: Native host→host, native VM→VM
- Also relevant: CPU usage on host (hardware emulation/serving) and VM.

Disk Transfer Rates (cat)

	NAT	Bridged	Mac
Native	31 MiB/s	31 MiB/s	272 MiB/s
HGFS	**35 MiB/s**	**43 MiB/s**	N/A
NFS	17 MiB/s	8 MiB/s	N/A
xrootd (cp)	20 MiB/s	tbd	70 MiB/s

No Caching (flushed RAM with big file)

Caching Effects (cat)

	Try 1	Try 2	Try3
Native VM	31 MiB/s	462 MiB/s	493 MiB/s
Native Mac (Large)	272 MiB/s	273 MiB/s	272 MiB/s
Native Mac	272 MiB/s	1707 MiB/s	1704 MiB/s
NFS (NAT)	17 MiB/s	46 MiB/s	38 MiB/s
NFS (Brid.)	8 MiB/s	Skipped - pointless	
HGFS (NAT)	**35 MiB/s**	43 MiB/s	41 MiB/s
HGFS (Brid.)	43 MiB/s	53 MiB/s	54 MiB/s

CPU % Load - 8 Core (cat)

	NAT	Bridged	Mac
Native	10%		40%
HGFS	**Inconstant: 24%**	Inconstant:	**120%**
NFS	60%		160%

Comments

- HGFS gives best performance... (even better than native VM disk!)
  - Not sure how it works... doesn't seem to use network (good).
  - But has some problems.
    - Misreports number of blocks in files!
    - Seems to have issues with symlinks.
      - Currently cannot install ATLAS kit in HGFS volume.
      - Need interaction with vmware.
- There seems to be lots of room for tuning in xrootd.
  - 70 MiB/s host → host should be reproducible host → VM
  - But this can't serve as a solution for home disk/code.
  - ROOT 5.19 (host) → 5.18 (VM) was 50% slower!
- Network adapter / NAT emulation ultimately steals cycles from VM.
- Scaling tests to > 1 VM per host now...

Other VM Issues

- VM/Host configuration:
  - VMs require optimized kernels.
    - eg: idle VMs host-CPU usage large due to frequent polling of the virtualized clock. Reduction idle CPU usage from 30% to 5% with proper kernel tuning.
    - Ideally these issues will be addressed by the CERN VM team.
  - SLC4.5 installation is easy... but again we would prefer a working/tuned OS installation from CERN.
  - Some things may be difficult to get working "out of the box".
    - Host (or servers) must serve disks... VMs must see them.
    - Batch queues.
  - Have had some issues with VOMS with VMs behind NAT (but OK with bridged network).

Building VMs for Experiments

- What is the model of interplay experiments and CERN VM team?
- There are VM-specific issues (kernel, OS, etc): ideally these will be handled by the CERN VM team
- There are experiment specific issues (SW, libraries, etc). Either the CERN VM team
  - provides mechanism to drop-in external components (eg disk)
  - experiment builds custom VMs based on VM from CERN team.
    - → There needs to be team on experiment side.
- There are host configuration issues (eg shared disk):
  - Installation scripts customized for host platform/role?
- Distribution: VMs are not small.
  - Experiment specific distribution?
  - Use: Web servers, GRID FTP, experiment DDM, torrents, DVDs, ... ?

Summary

- There is a lot of potential for using VMs in ATLAS.
- Ultimately the benefit of VM is the ease of setup/use for non-experts.
  - To achieve this, an experiment-specific component is necessary.
- Our first focus is using VMs to make it easy for users to do analysis on laptops/home institutions.
  - Nearly ready to distribute our first attempt to users.
  - Storage is an important element here... we are trying to figure out what is the best configuration.
  - Next step is to scale to lots of VMs on various machines.
    - Building a tier 3 from leveraged resources.
- We would greatly benefit from integrating with CERN VM team activities... we just need to understand what this means.