Managing Virtual Machine Images

2 November 2007

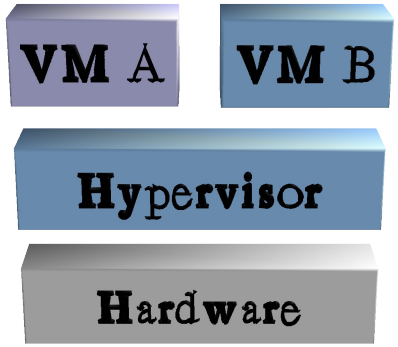
Håvard Bjerke

Agenda

- HW virtualization introduction and motivation
- OS Farm
  - tool for creating and storing VM images
- Content Based Transfer
  - technique for efficient transfer of VM images

HW virtualization intro

- Allows running several virtual machines (VMs) simultaneously on a single physical machine
- Classic consolidation scenario:
  - Run database and web server on the same machine
  - Run services in separate VMs, pinned to separate CPU cores
  - Save $



HW virtualization intro

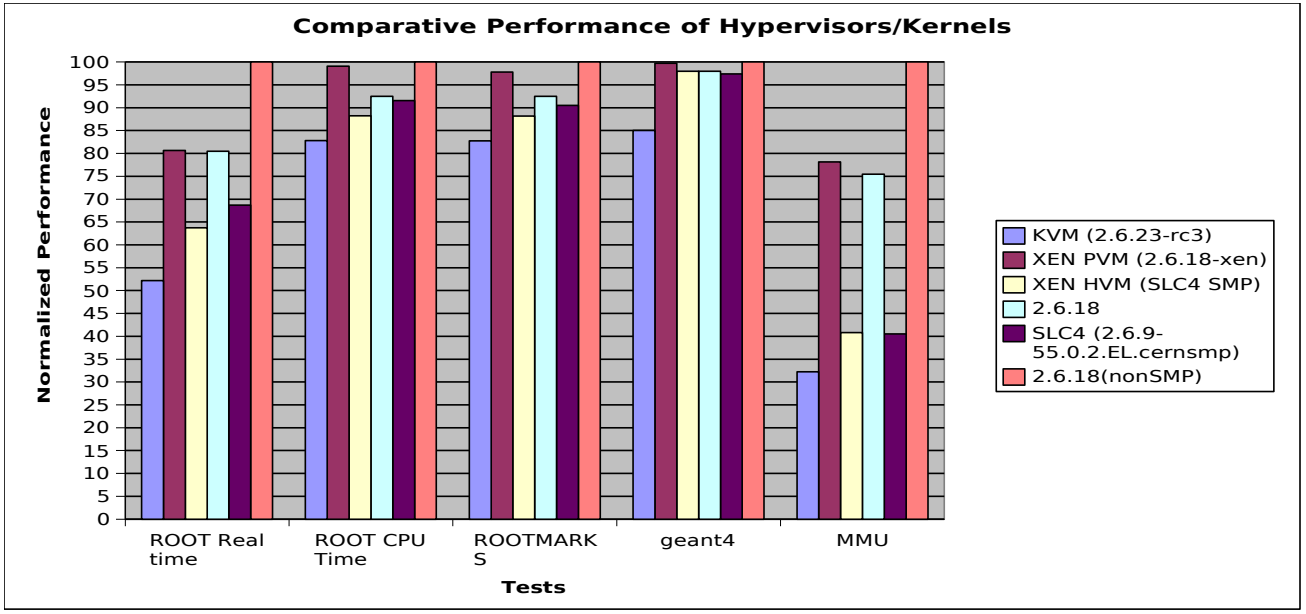
- Benefits for Grid
  - Secure isolation
    - Small Trusted Computing Base in Xen
    - Isolate malicious software
  - Software flexibility
    - Better ability to satisfy requirements for execution environments
    - E.g. run both SLC3 and SLC4 on one physical node
  - Serialization, Live migration
    - Migrate essential services upon
      - hardware failure, or
      - maintenance

Recent developments

- HW acceleration
  - 1st generation Intel VTx CPU extensions
    - Allow full virtualization without binary rewriting or interpretation
    - A -1 or VMX privilege level
    - Already mainstream in Core architecture
  - 2nd generation Intel VTx CPU extensions
    - Add Extended Page Tables
    - Support guest VMs' page tables nested inside host's page tables
  - Intel VTd chipset extensions allow more efficient partitioning of I/O
    - Allocate device addresses to VMs

Virtualization at CERN

- Xen's virtual hardware has proved itself to be a good competitor to physical hardware
- Adds convenience while negligibly affecting performance



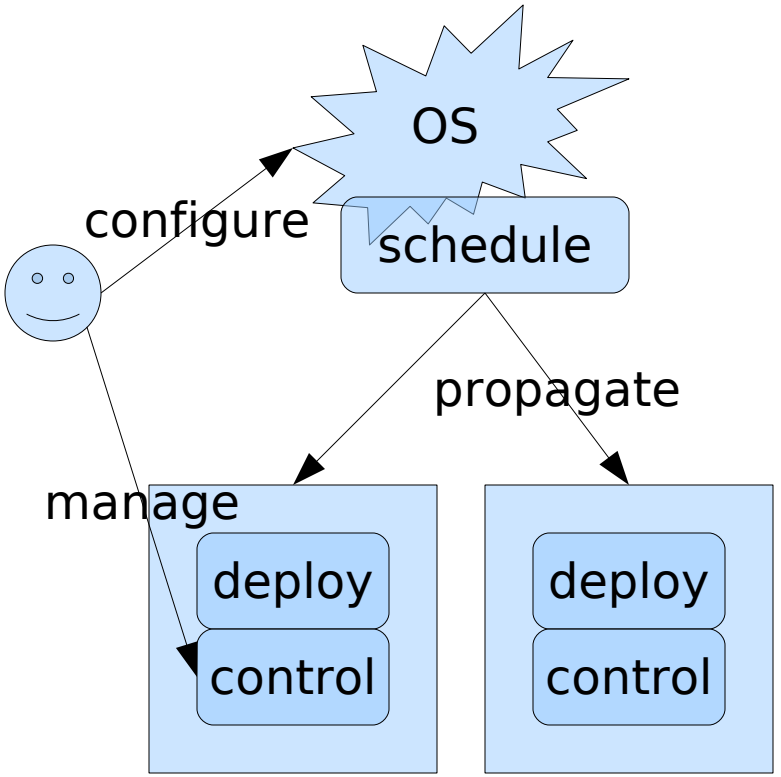
Virtualization trends

- Perhaps most significant (Grid related) developments

Past	Present	Future
•VM technology	•VM deployment •Virtual appliances •HW acceleration •Testing environments	•Authorization •Security •Contextualization •Resource negotiation

Virtual Machine Management

- Offline
  - VM generation
  - VM scheduling
  - VM propagation
  - VM deployment
- Online
  - VM management
- Both
  - VM configuration



VM deployment

- Two tools already developed at CERN
  - SmartDomains
    - life-cycle management
  - vGrid
    - portal based
- Other models
  - Virtual Workspaces
    - VM scheduling and propagation
  - Batch system customization
    - LSF
    - Torque / MOAB scheduler

OS Farm

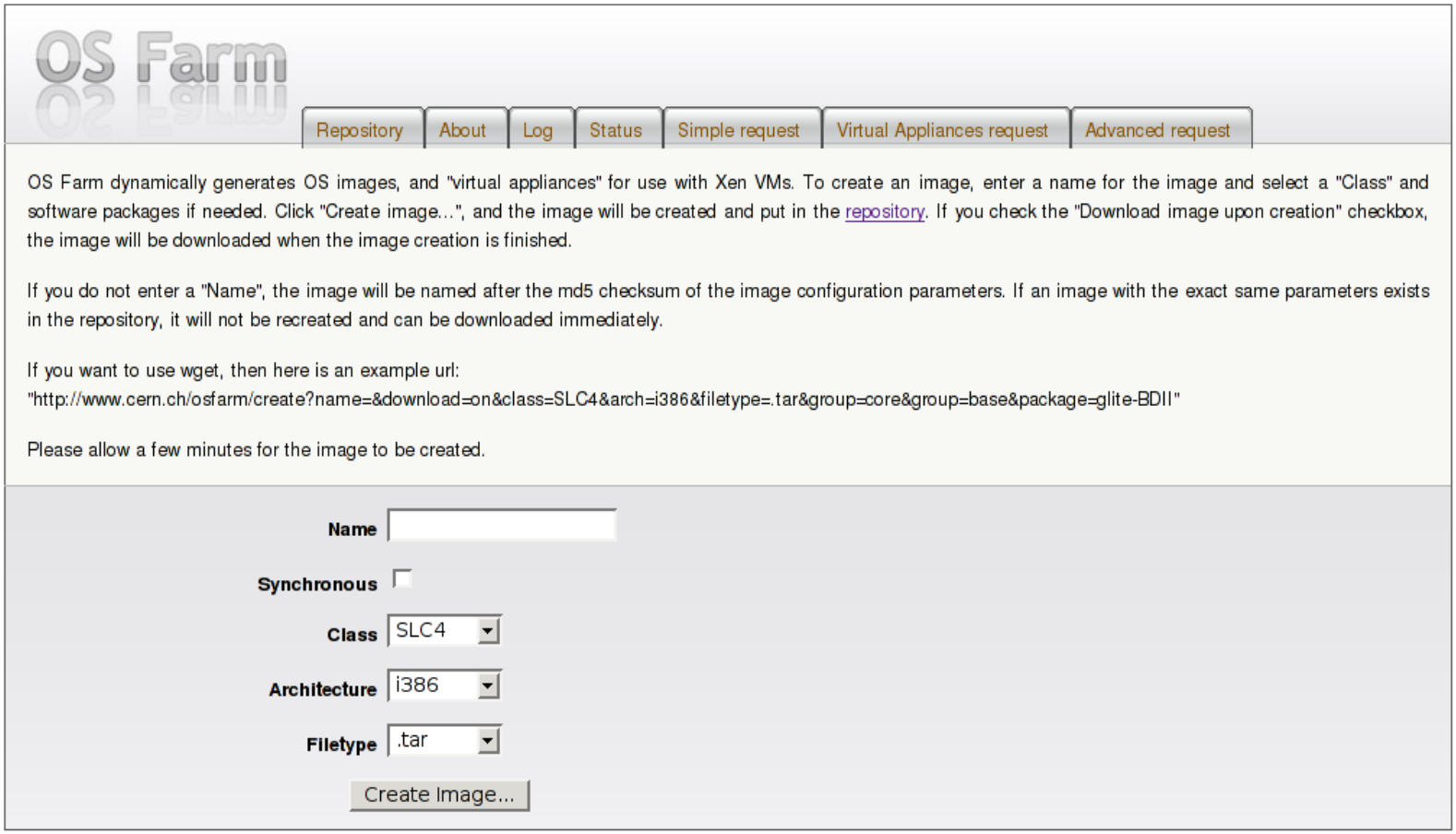
OS images

- Various projects at CERN need quick deployment of clean machines
  - Software building and testing
- Need a rich set of Linux flavours
  - The default Linux distribution for CERN is Scientific Linux CERN 4 (SLC4)
    - SLC3 is being replaced by SLC4, but some software still depends on SLC3
    - SLC4 will be replaced by SLC5
  - Other flavours are needed for testing: Debian, Ubuntu, CentOS, Scientific Linux
- Windows, Solaris

OS images

- In a batch system, VM images can be used to satisfy execution environment requirements
  - A static set of requirements can be satisfied easily with a library of images
  - But requirements can have combinatorial complexity
  - Ultimately, users supplying their own images provides best coverage
  - Also better for software licenses

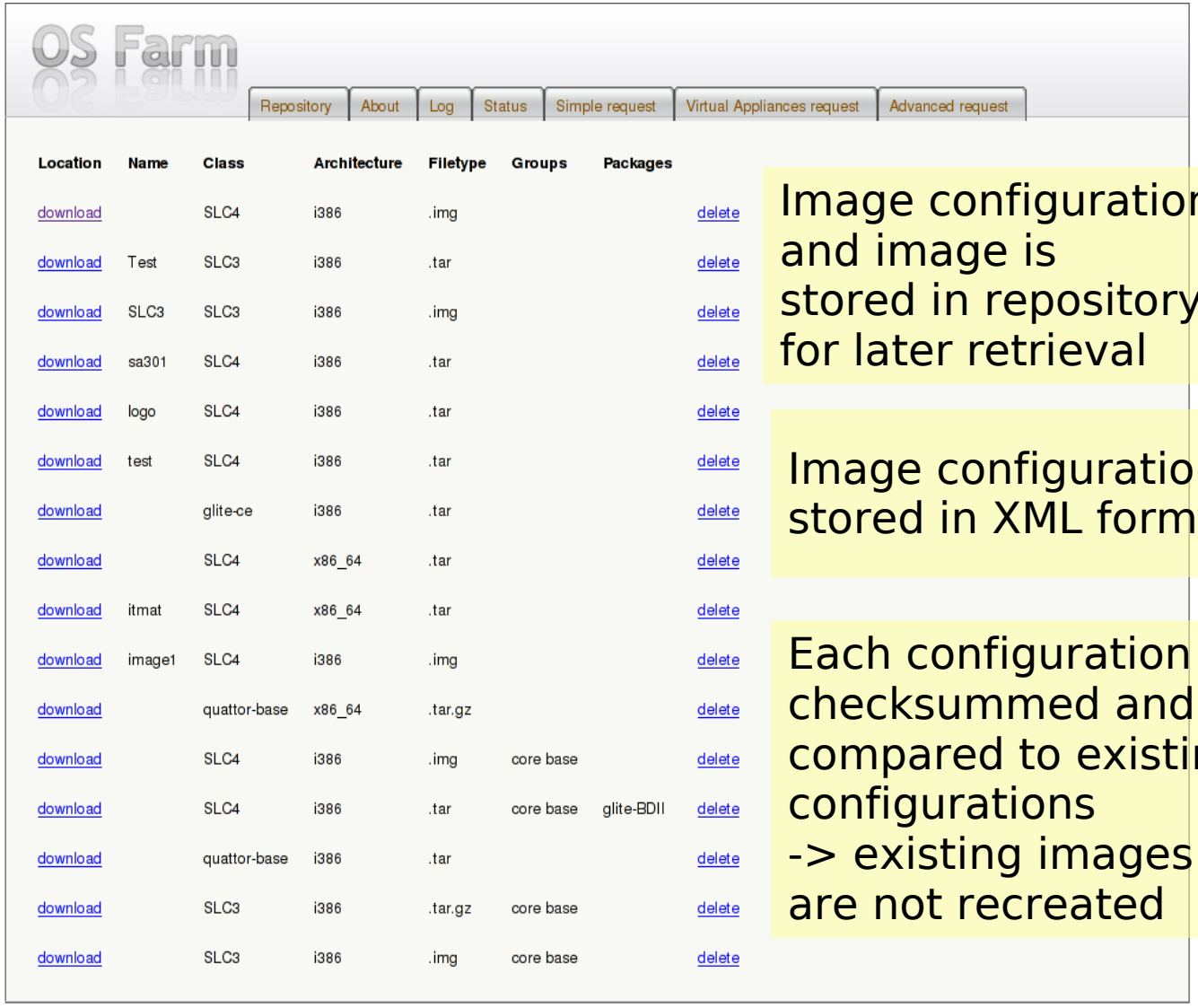
Complementary tool: OS Farm



Flavours

- Base images
  - SLC 3 & 4 – standard at CERN
  - libfsimage – basis for several flavours
    - Debian and Red Hat based distributions
- Virtual appliances
  - gLite - Grid middleware
    - gLite-CE
    - gLite-WN
  - Quattor - fabric management
- 32 and 64 bit images
- tar or raw (*.img) image format

Repository



Location	Name	Class	Architecture	Filetype	Groups	Packages	
download		SLC4	i386	.img			delete
download	Test	SLC3	i386	.tar			delete
download	SLC3	SLC3	i386	.img			delete
download	sa301	SLC4	i386	.tar			delete
download	logo	SLC4	i386	.tar			delete
download	test	SLC4	i386	.tar			delete
download		glite-ce	i386	.tar			delete
download		SLC4	x86_64	.tar			delete
download	itmat	SLC4	x86_64	.tar			delete
download	image1	SLC4	i386	.img			delete
download		quattor-base	x86_64	.tar.gz			delete
download		SLC4	i386	.img	core base		delete
download		SLC4	i386	.tar	core base	glite-BDII	delete
download		quattor-base	i386	.tar			delete
download		SLC3	i386	.tar.gz	core base		delete
download		SLC3	i386	.img	core base		delete

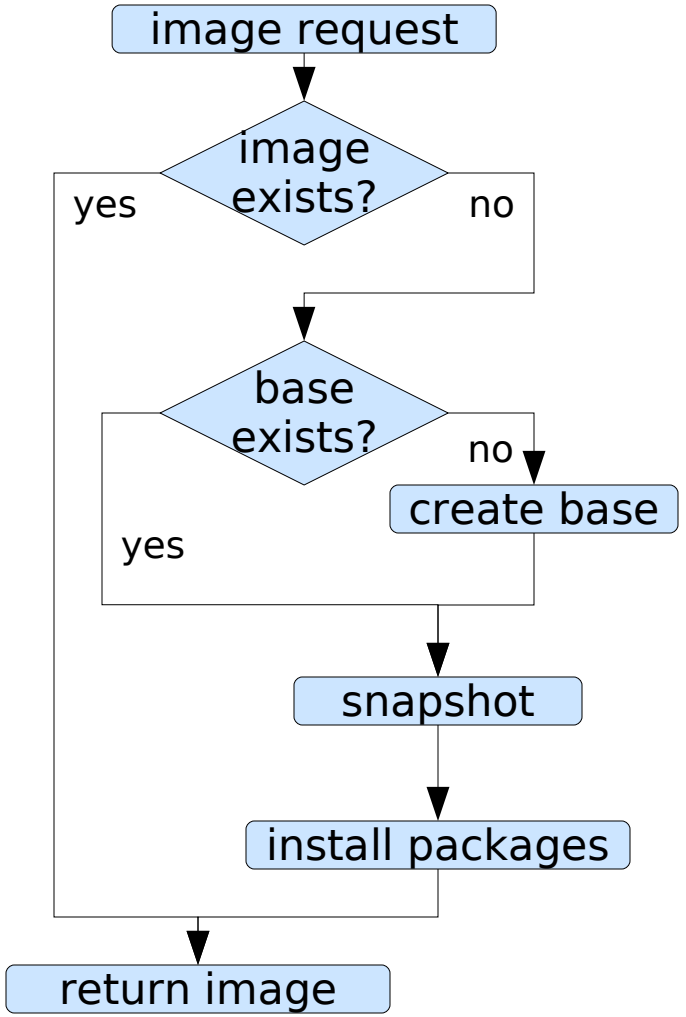
Image configuration and image is stored in repository for later retrieval

Image configuration is stored in XML format

Each configuration is checksummed and compared to existing configurations
-> existing images are not recreated

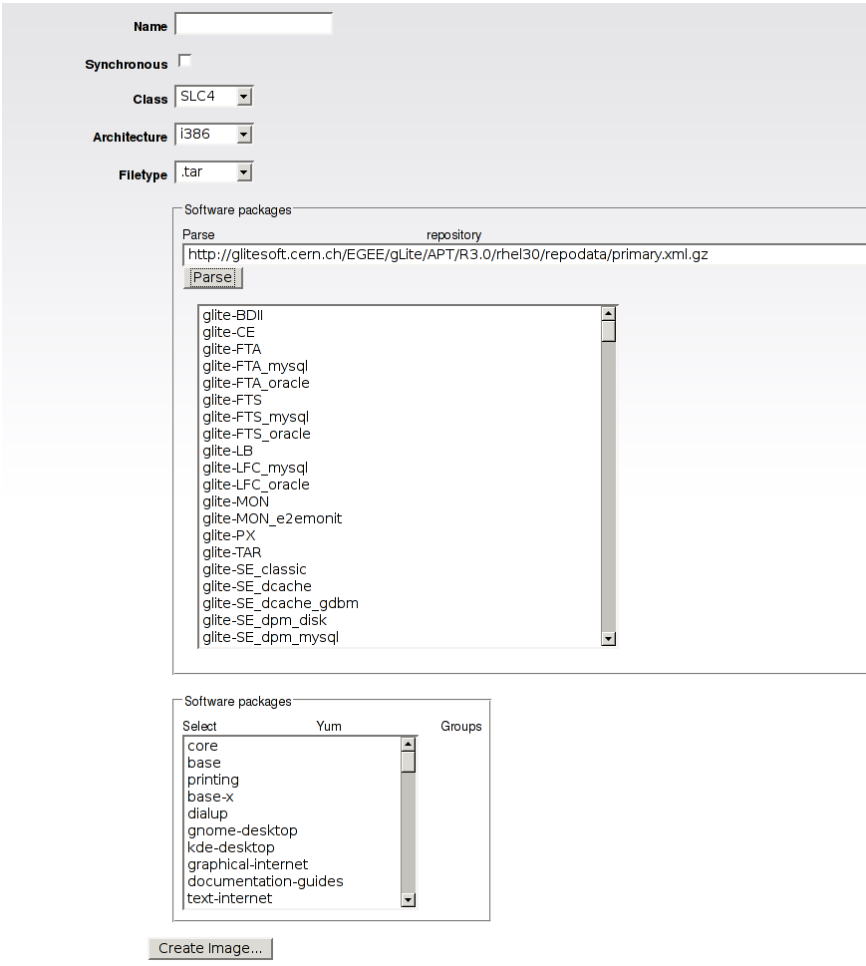
Copy-on-Write staging

- Images are created dynamically
- Base stages are kept in cache
- Uses LVM snapshots (copy-on-write) for instantaneous staging



Parse yum repositories

- Most gLite and CERN fabric software reside in yum repositories
- Easy to add yum packages from yum repositories
- Supports most of CERNs fabric packages



Compared to rPath

- rPath – virtual appliance builder
  - Provides a massive selection of virtual appliances
  - Supports many different image formats
  - Roles separated into image authors and users
  - Proprietary build software
- OS Farm
  - Only one role – the user or a computer program
  - Dynamic image requests – image generated on the fly
  - Fully open source

Image security

- The Xen hypervisor isolates VMs from each other
  - As secure as or more secure than processes running in different address spaces
  - Xen has a small Trusted Computing Base
- If the user creates the VM image, she can have root privileges in the VM
  - Convenient for the user
  - Is it safe?
    - Promiscuous network access
  - Violations are traceable (encapsulation)

Image signing

- Still, most administrators would be reluctant to allow users to deploy their own VM images on the grid
- Great effort goes into certifying SLC
- Running an uncertified OS on CERN Computer Centre hardware is risky
- Running an uncertified OS in an isolated VM is less risky

Image signing

- Solution: Secure images
  - Controlled by admin (in control of propagation and deployment)
    - Administrator provides a static library of trusted images
    - Administrator provides a trusted image generation service
  - Trusted by certification, signed by
    - people
    - trusted image generation service

Contextualization

- There are several different OS image formats between vendors
  - VMWare (proprietary format)
  - Xen PVM
  - Xen HVM, KVM
- Images need host and site-specific configuration (network, etc.)
  - Images can be configured at deployment (parameterized) or pre-configured
- Goal: run my image on any Grid (LCG, Amazon EC2, my laptop)

Open Virtual Machine Format

- A DMTF (Distributed Management Task Force) Specification
  - Vendor neutral
  - Metadata specification
    - Virtual hardware description (CIM based)
    - CPU compatibility section
    - Feedback protocol
    - Service level requirements
  - Manifest
  - Certificate

Current & future developments

- SOAP interface
  - Platform and language independent interface
- XML image descriptions
- Signed images (verify integrity)
- Content based image transfers

Content Based Transfer

Image staging

- Jobs in the queue normally assume that files already exist at the execution node (e.g. through NFS) or that they will be staged when the job starts
- But VM images are big
  - ~ 300 MB to several GB
  - Jobs will have to wait for the image transfer to finish
  - Congests network

Content-Based Addressing

- Possible solutions?
  - Image cache
  - Transfer the image before the job is scheduled – image has to be on each possible target node
- Hard to satisfy each possible image configuration
- Observation from Content-based Addressing
  - Most images are relatively similar
  - No need to transfer the whole image; just transfer the delta

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

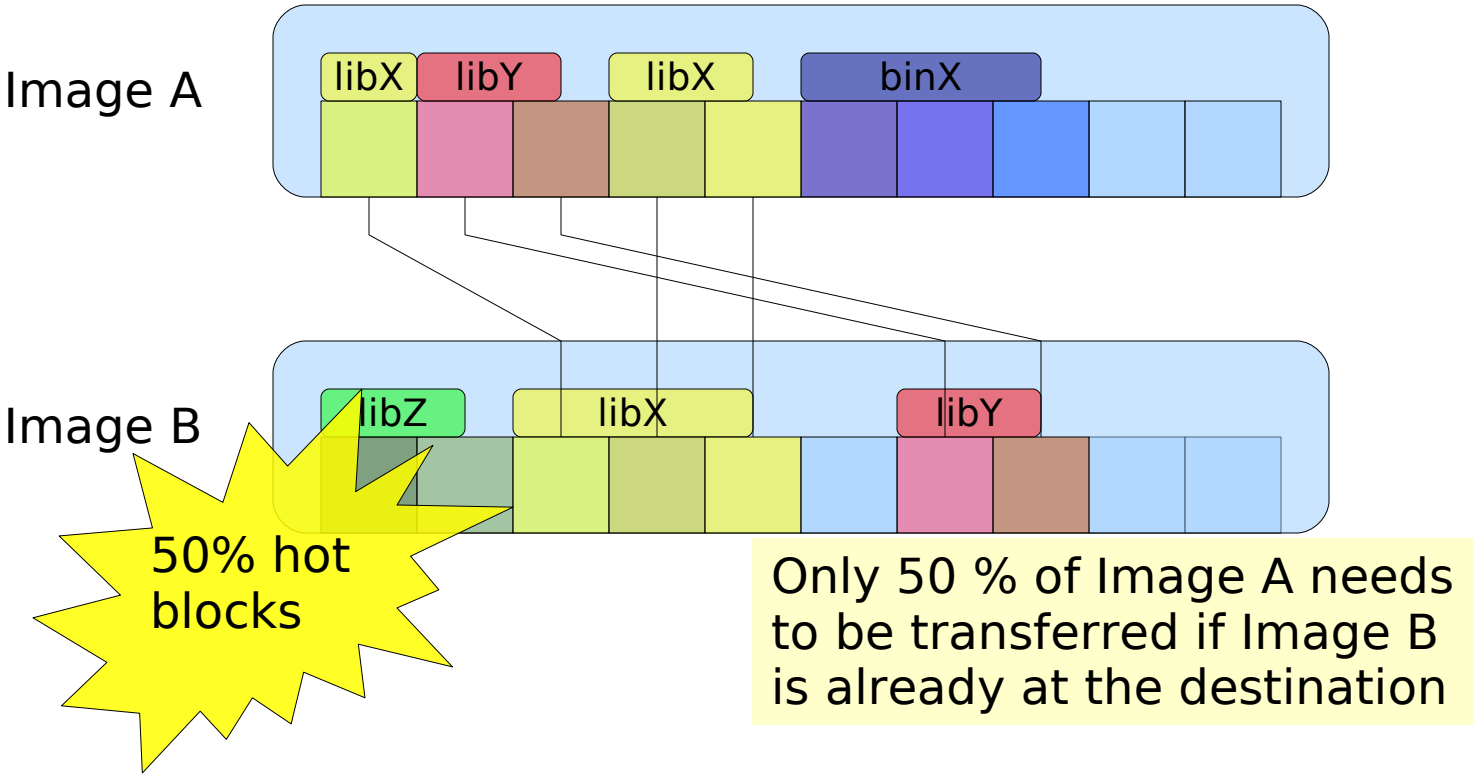
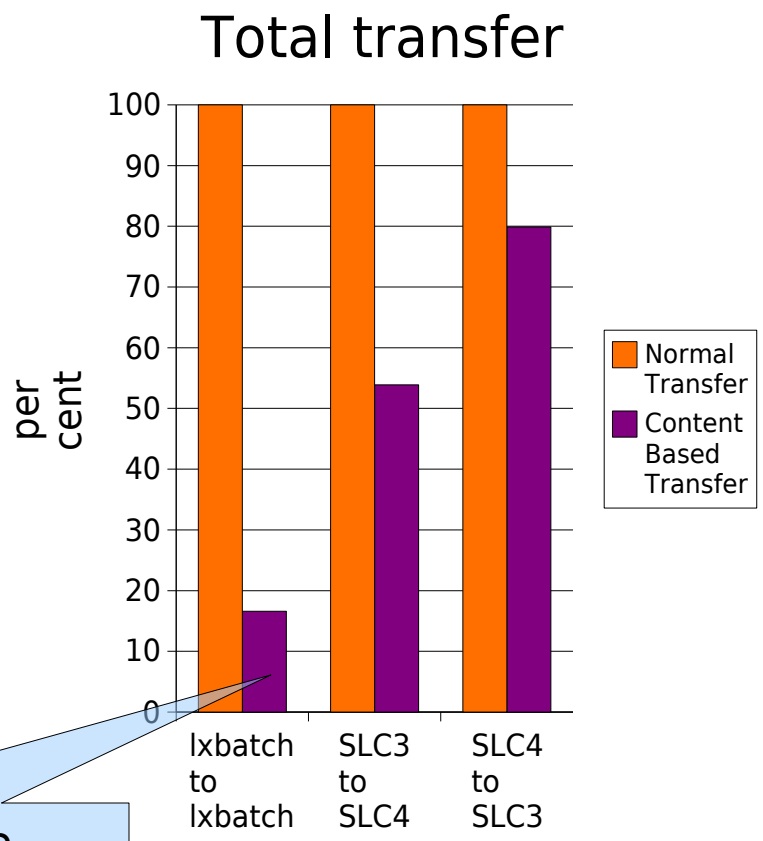


Image comparisons

- Two typical batch machines (5.3 GB)
  - 84 % hot blocks
- SLC3 (343 MB) and SLC4 (762 MB)
  - SLC3 -> SLC4
    - 48 % hot blocks
  - SLC4 -> SLC3
    - 22 % hot blocks

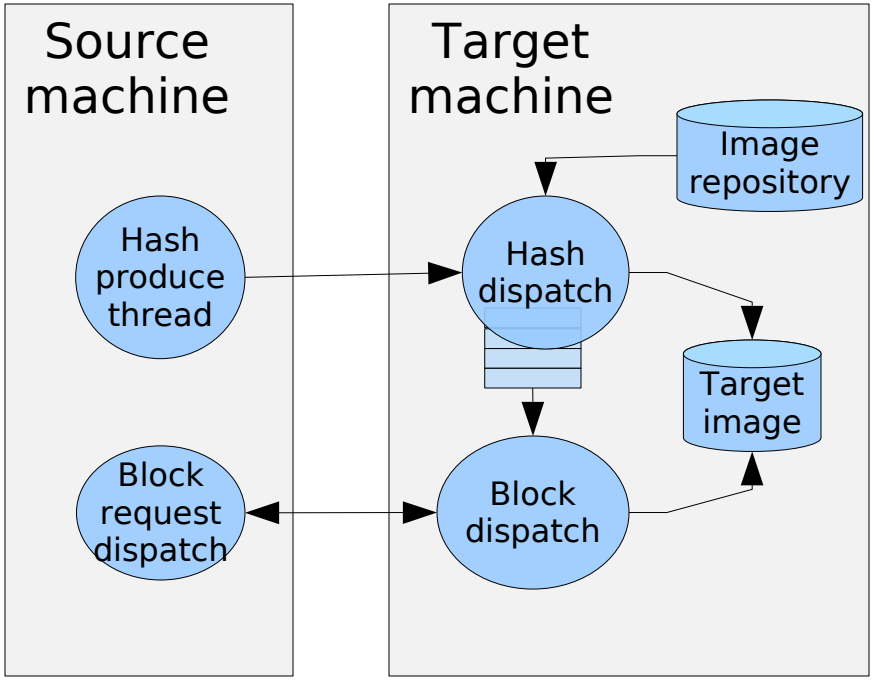


Fraction of full image data needed to transfer, including hash table

Cost

- Generating hash tables for source file and target repository
  - Linear
- Accessing hash tables
  - Java and Python have convenient constant-time hash tables
- Hash table data overhead
  - Depends on
    - hash function, e.g. SHA is 20 bytes
    - block size – usually 4096 bytes
  - 0.48 to 2.0 % of the image size

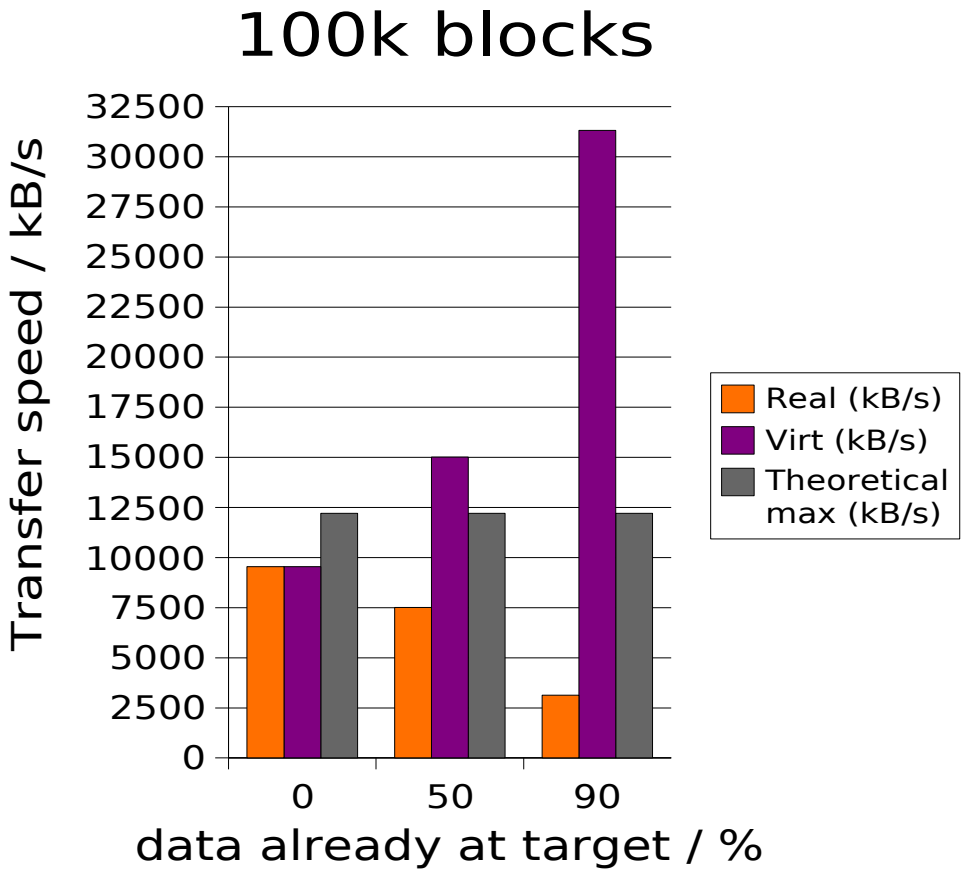
Content Based Transfer



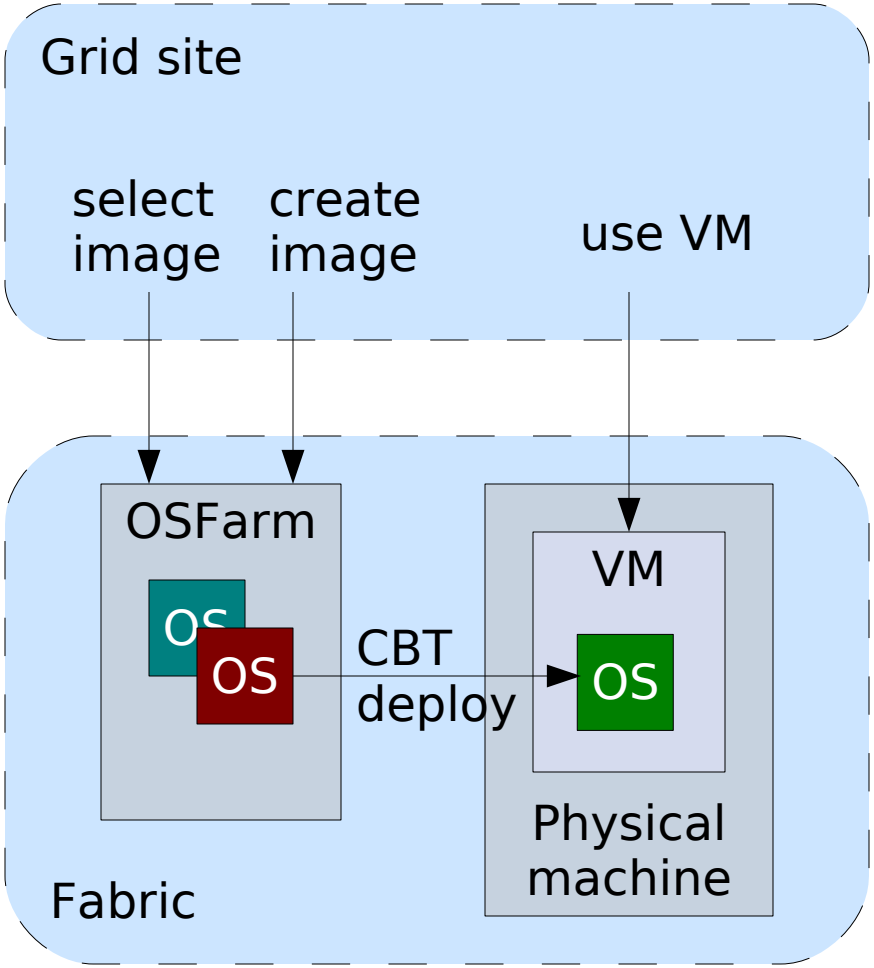
- Multithreaded
- Hash calculation and data transfer pipelined
- Implemented in Java (+ a Python prototype)

Measurements of CBT tool

Virtual speed:
full image size /
time to transfer delta



In control over Grid VM images



More information & Questions

- OS Farm
  - http://cern.ch/osfarm
- Content Based Transfer
  - http://hbjerke.web.cern.ch/hbjerke/cba/cba.xml