Virtualization

15 November 2007

Håvard Bjerke

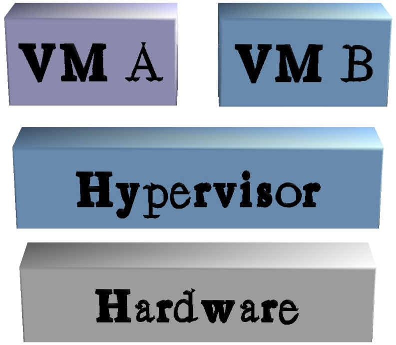
CERN openlab

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 different services in separate VMs, pinned to separate CPU cores
  - Save $



Virtualization in Grids

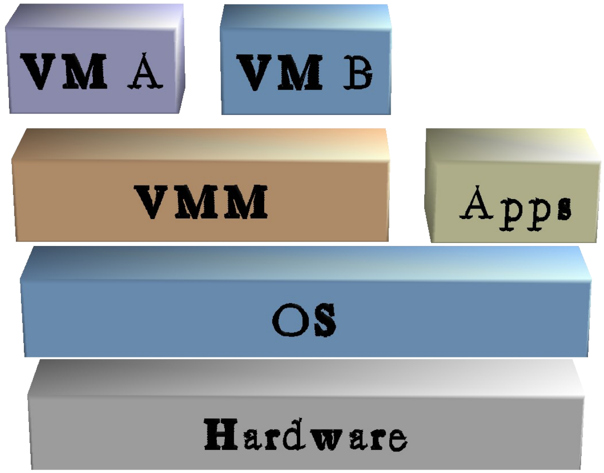
- Benefits for Grids
  - Secure isolation
    - Small Trusted Computing Base in Xen
    - Isolate malicious software
  - Software flexibility
    - Better ability to satisfy requirements for execution environments
    - E.g. run both SLC3 and SLC4 on one physical node
  - Serialization, Live migration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

Virtualization Attributes

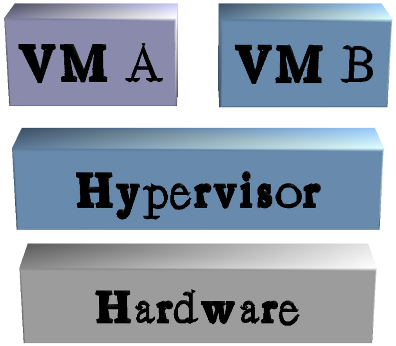
- Hosted vs non-hosted models
- Technique
  - Paravirtualization vs full virtualization
  - Binary rewriting
- Hardware acceleration
  - Intel VT CPU and chipset hardware extensions
- Performance attributes
  - I/O performance
  - CPU performance

Virtualization Models

- Hosted
  - VMWare Server
  - Microsoft Virtualization Server



- Non-hosted
  - Xen
  - VMWare ESX



Virtualization Techniques

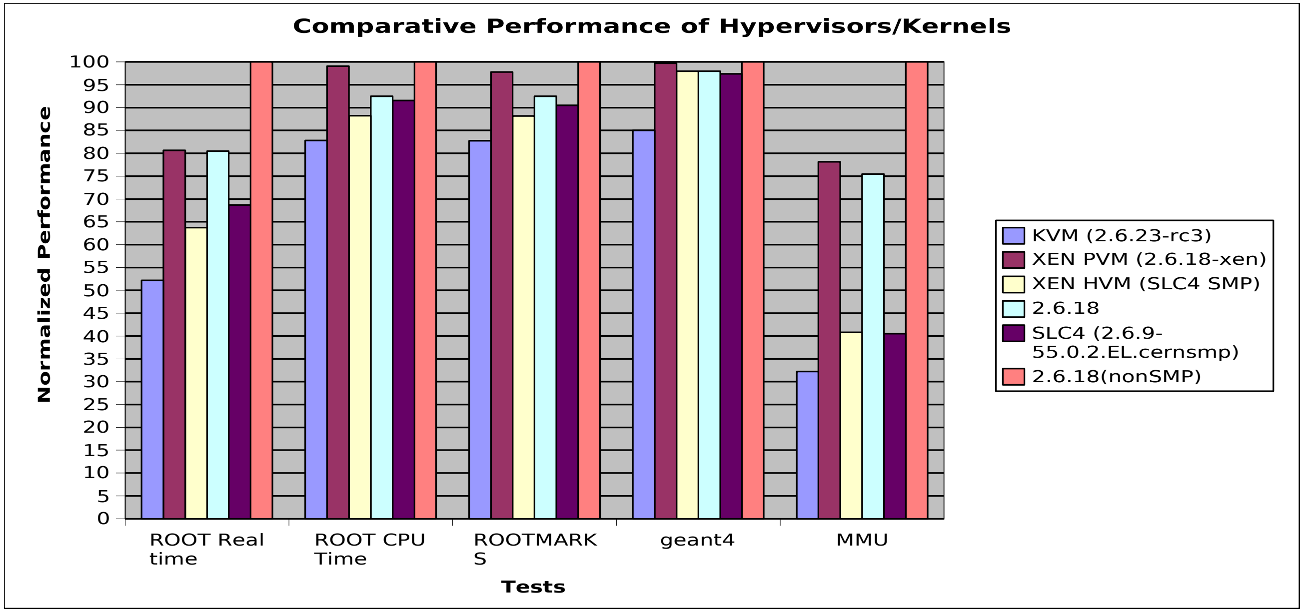
- Paravirtualization
  - Requires cooperation from guest operating system
  - Requires modification to source code of guest OS
    - Linux, Solaris and FreeBSD is OK
    - MS Windows not OK
  - Examples: Xen, lguest
- Binary rewriting/patching
  - Guest OS execution is modified at runtime
  - Does not require modification of guest OS
    - MS Windows, Linux, etc. OK
  - Examples: VMWare Server, MS Virtualization Server

Hardware Acceleration

- 1st generation Intel VTx CPU extensions
  - Allow full virtualization without binary rewriting or interpretation
  - A -1 or “VMX Root” 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

Performance Attributes

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



Virtualization Landscape

	Non-hosted	Hosted
Binary rewriting	VMWare ESX	VMWare Server MS Virtualization Server
Paravirtualization	Xen PVM	lguest
HW assisted	Xen HVM	KVM

VM deployment

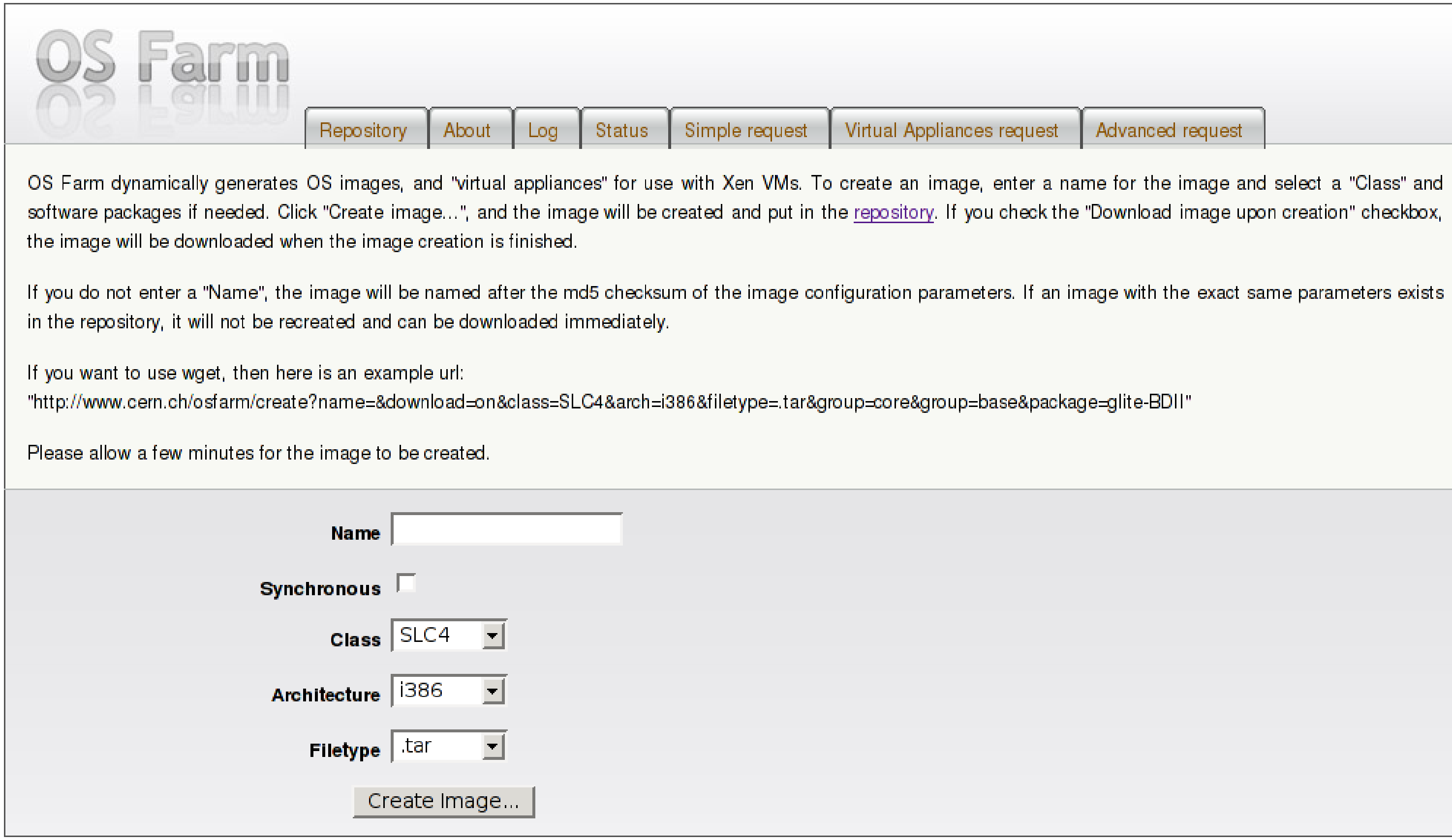
- Two tools already developed at CERN
  - SmartDomains
    - life-cycle management
  - vGrid
    - portal based
- Other models
  - Intel Grid Programming Environment (GPE)
  - Virtual Workspaces
    - VM scheduling and propagation
  - Batch system customization
    - LSF
    - Torque / MOAB scheduler

Grid Programming Environment

- Easy to develop and deploy Grid application beans
- Service-oriented Architecture
  - Target systems
  - Job management
  - Storage management
  - File transfer
- Uses virtual machines for resource provisioning
- The only Grid middleware to offer full platform virtualization support

Complementary tool: OS Farm

- Web interface:



- + SOAP web service interface

VM OS image Flavours

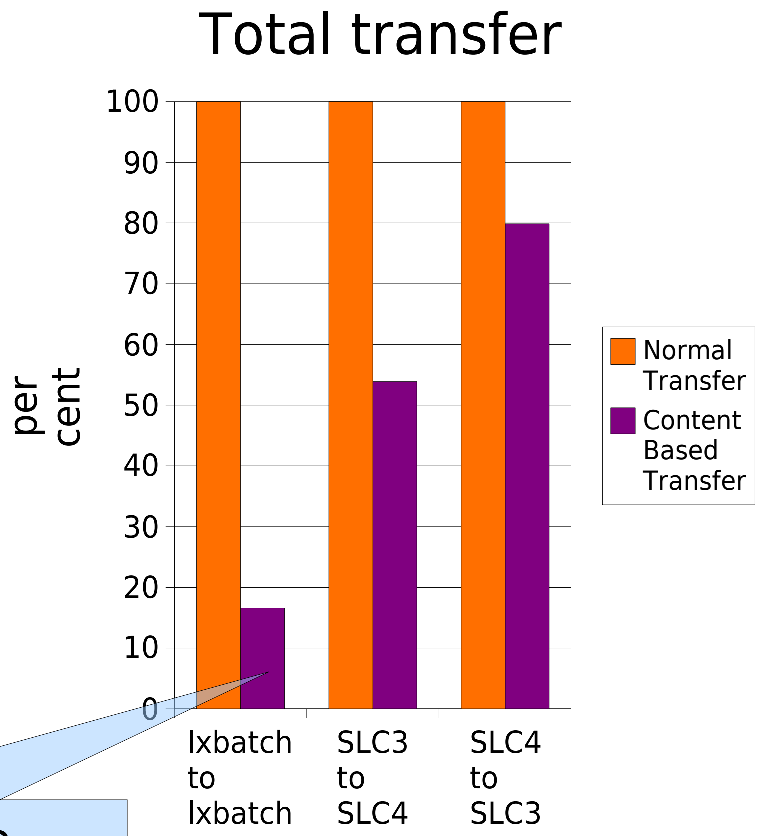
- Base images
  - Scientific Linux CERN 3 & 4 – standard at CERN
  - libfsimage – basis for several flavours
    - Debian and Red Hat based distributions
- Virtual appliances
  - gLite - Grid middleware
    - gLite-CE
    - gLite-WN
  - Quattor - fabric management
- 32 and 64 bit images
- tar or raw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

Image staging

- OS images are big
  - ~ 300 MB to several GB
  - Jobs scheduled for a VM will have to wait for the image transfer to finish
  - Congests network
- Observation from Content-based Addressing
  - Most images are relatively similar
  - No need to transfer the whole image; just transfer the delta

Image comparisons

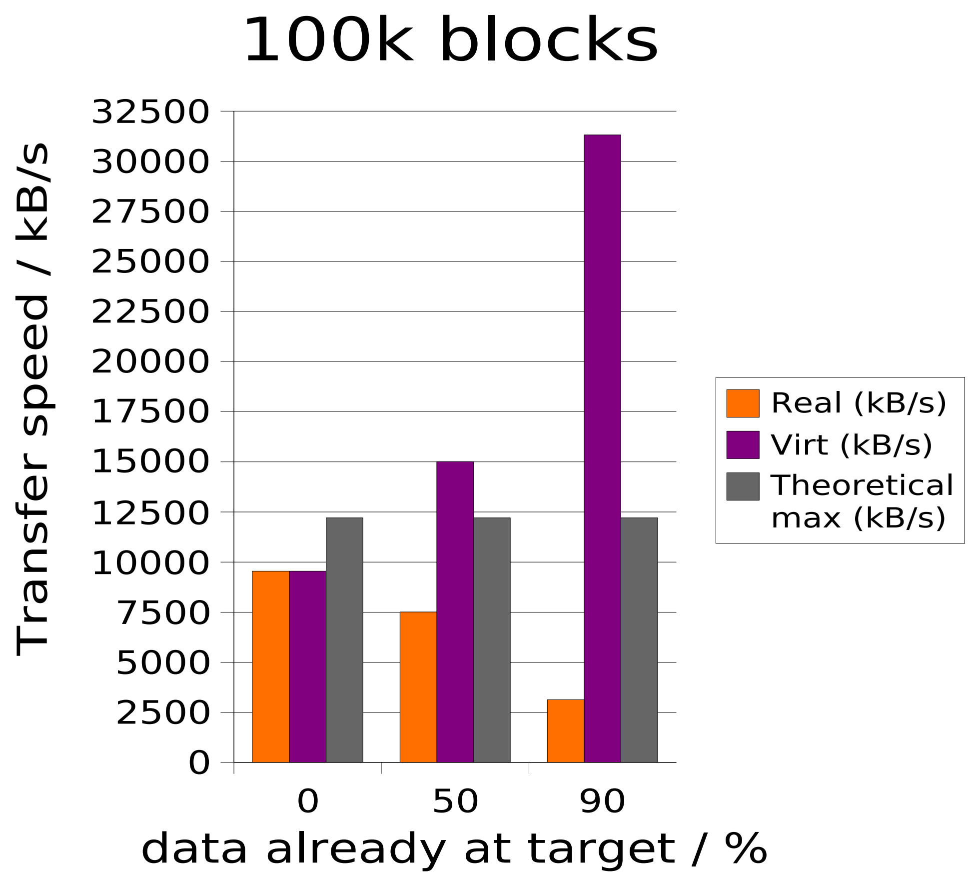
- Two typical batch machines (5.3 GB)
  - 84 % hot blocks
- SLC3 (343 MB) and SLC4 (762 MB)
  - SLC3 -> SLC4
    - 48 % hot blocks
  - SLC4 -> SLC3
    - 22 % hot blocks



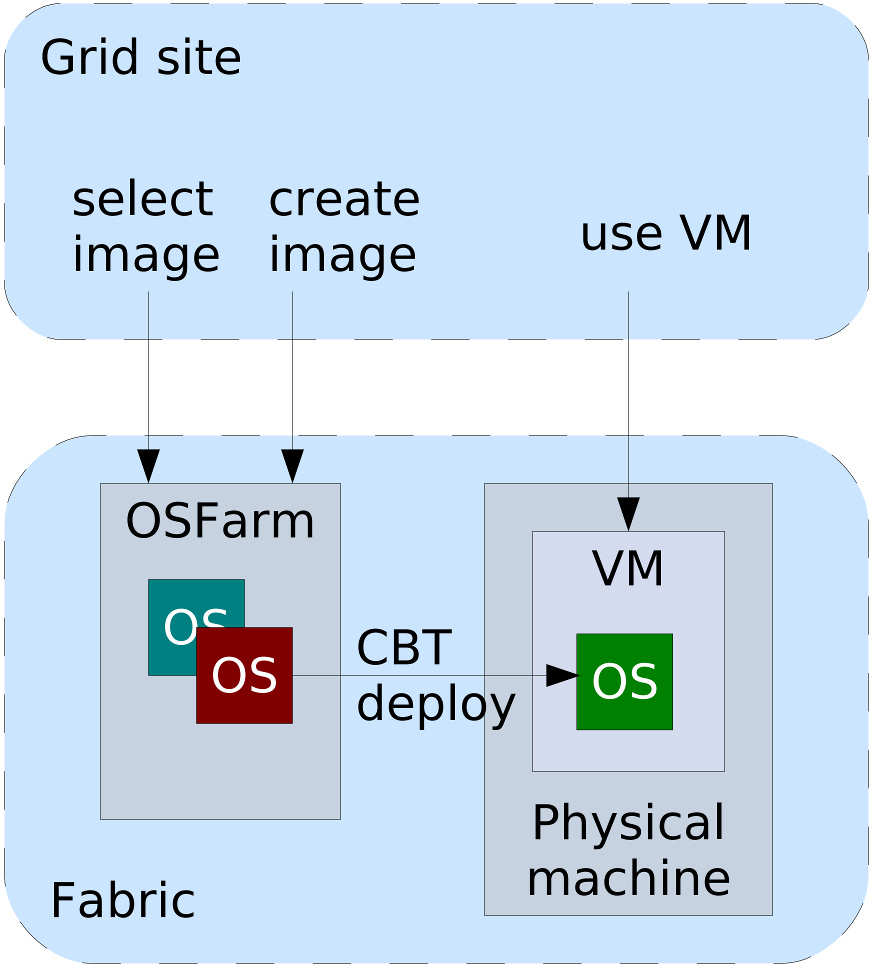
Fraction of full image data needed to transfer, including hash table

Measurements of CBT tool

Virtual speed:
full image size /
time to transfer delta



In control over Grid VM images



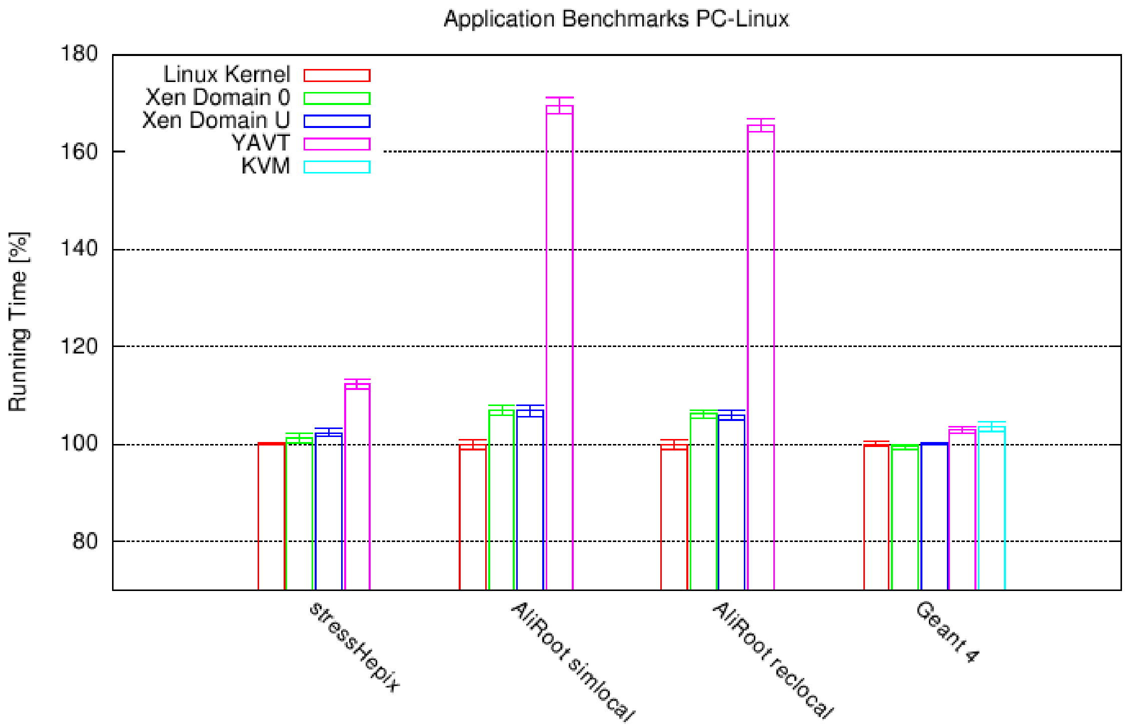
More information & Questions

- OS Farm
  - http://cern.ch/osfarm

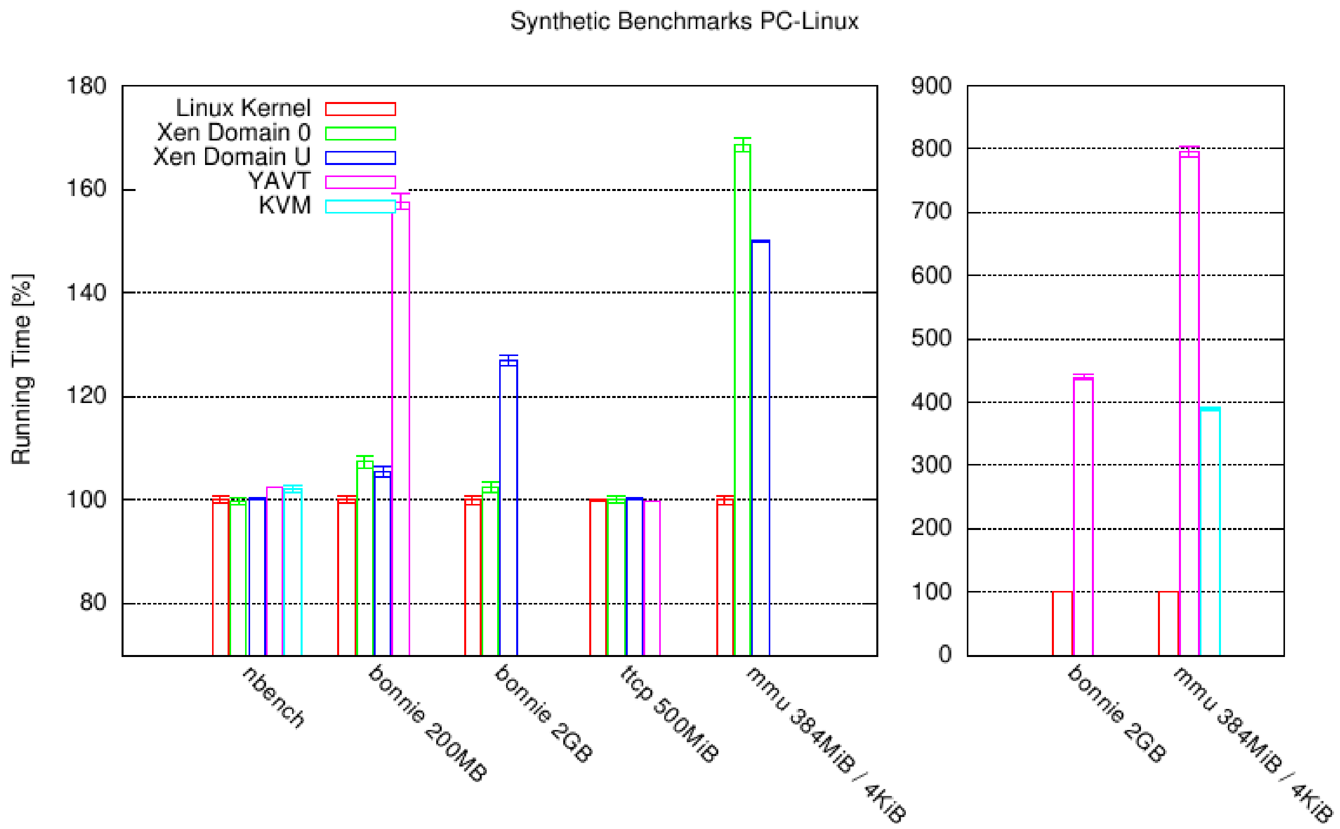
- Content Based Transfer
  - http://hbjerke.web.cern.ch/hbjerke/cba/cba.xml

Backup

More benchmarks

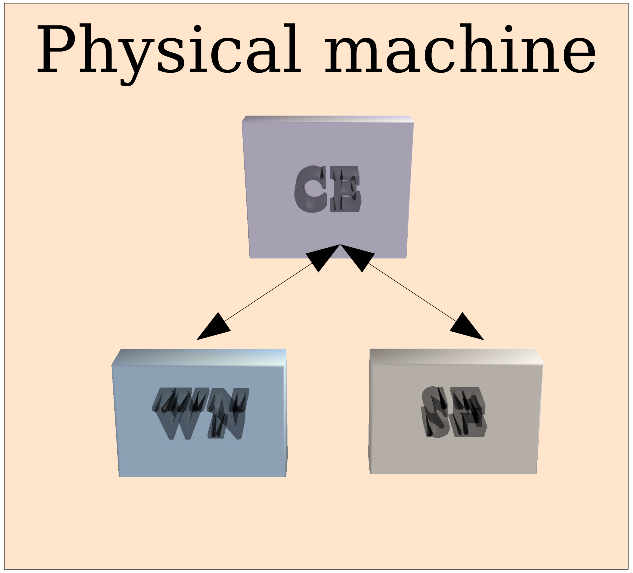


More benchmarks



Consolidation Example

- GRID-in-a-box
- Useful for testing or setting up proof of concept GRIDs
  - Regression testing
  - Network testing
  - Distributed application testing
  - Build testing



CERN virtualization use cases

- LCG
- Smartfrog (HP)
  - Utility computing
  - Single component description for a whole virtual cluster
  - Deploy a complete site – clean up afterwards
- Tycoon (HP)
- Health-e Child

Repository

OS Farm

Repository | About | Log | Status | Simple request | Virtual Appliances request | Advanced request

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

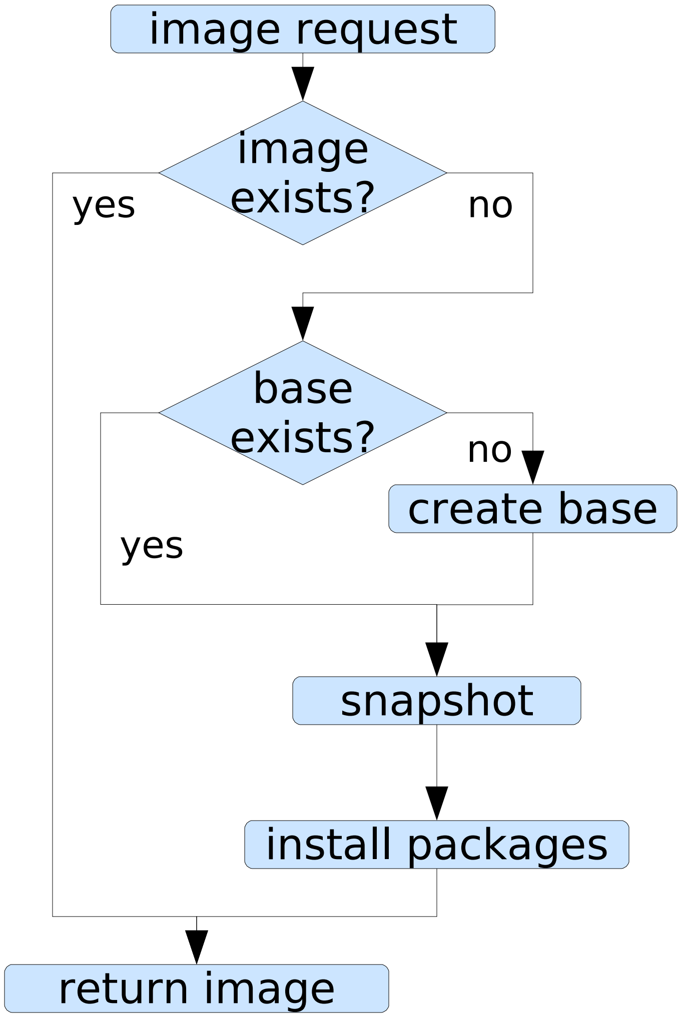
Image configuration and image is stored in repository for later retrieval

Image configuration is stored in XML format

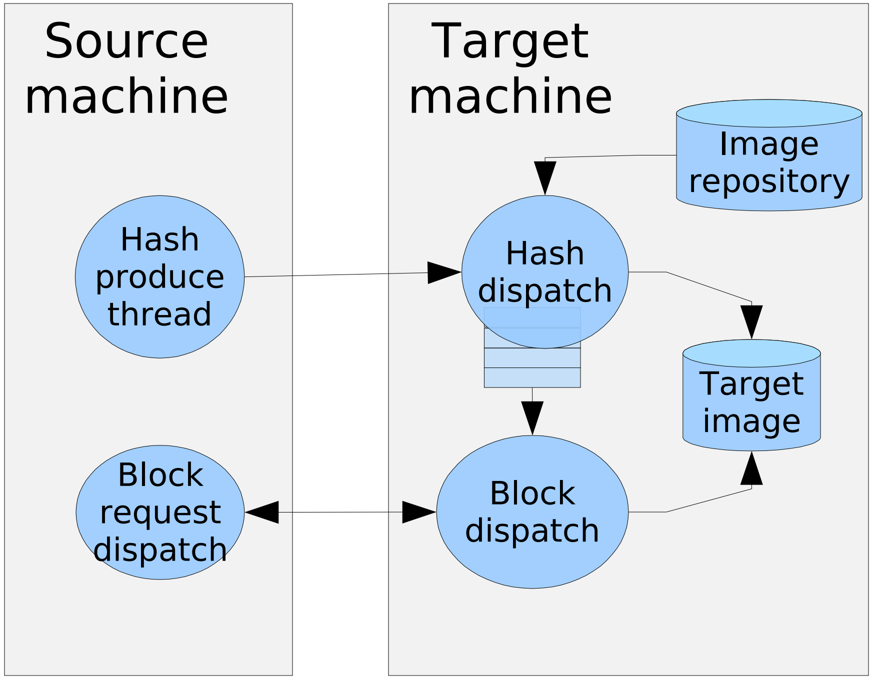
Each configuration is checksummed and compared to existing configurations
-> existing images are not recreated

Copy-on-Write staging

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



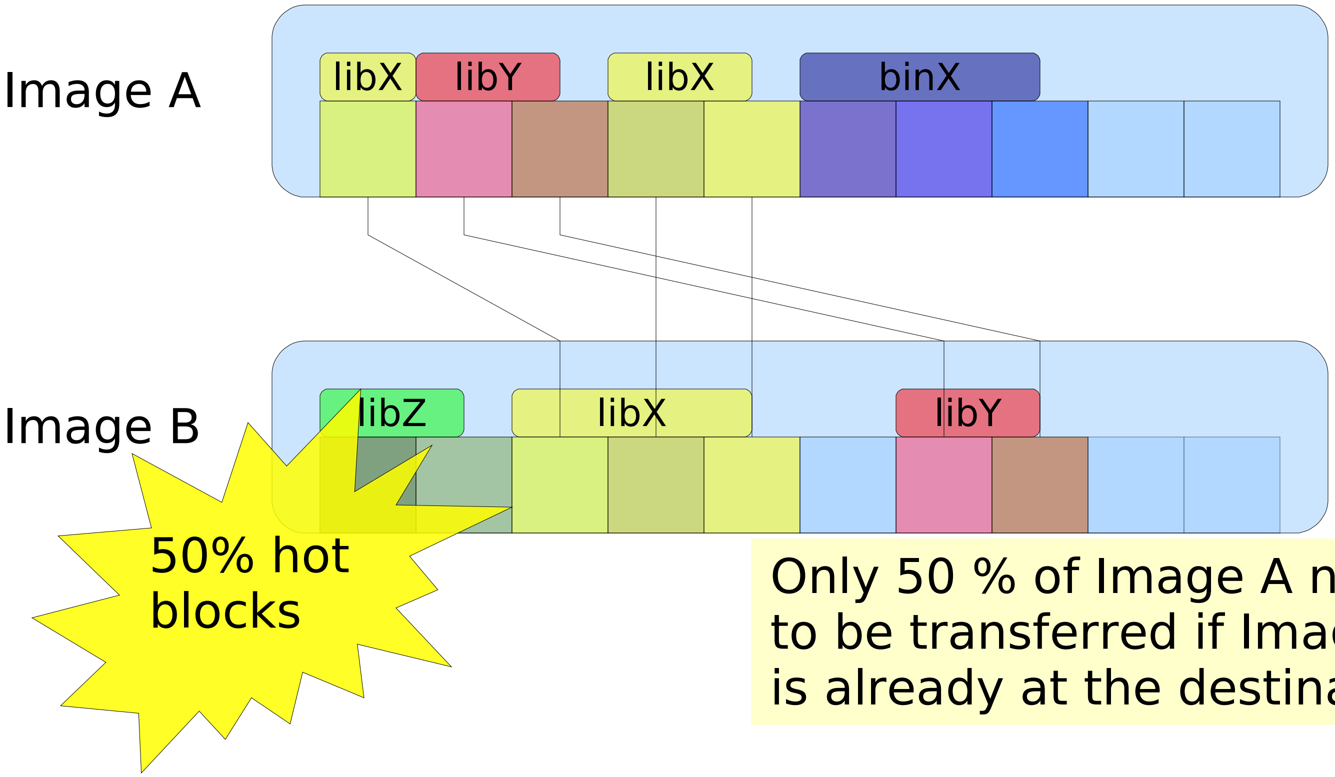
Content Based Transfer



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

Filesystems

- 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

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