Process Migration using Virtual Machines

19 June 2007

Håvard Bjerke

Why migrate processes?

- Better and more flexible use of resources
- Migrate *away* from poor or broken resources
  - Provide for easier server maintenance
- Migrate *towards* more suitable resources
- Load leveling
- Optimize throughput

Checkpointing with VMs

- Time segmenting
  - Divide the execution of a job into time segments
  - The failure of one segment does not fail the whole job
  - Prevent failure of long-hauled jobs
    - Run infinitely
  - Thus, the execution node does not need to give any guarantees for the whole job
- Multitasking
  - Preemption for higher-priority jobs

VM migration in the Grid

- Why use VMs for migration in the Grid?
  - Submit execution environment with job
  - Eliminate software matchmaking
  - Avoid software related “black holes”

Black holes: nodes that advertise resources incorrectly and continuously attract jobs that fail because of missing resources

  - Run anywhere
    - No grid software needs installing on computing node
    - Suitable for public computing
  - Independence from specific grid software
    - Run globally, across incompatible grids

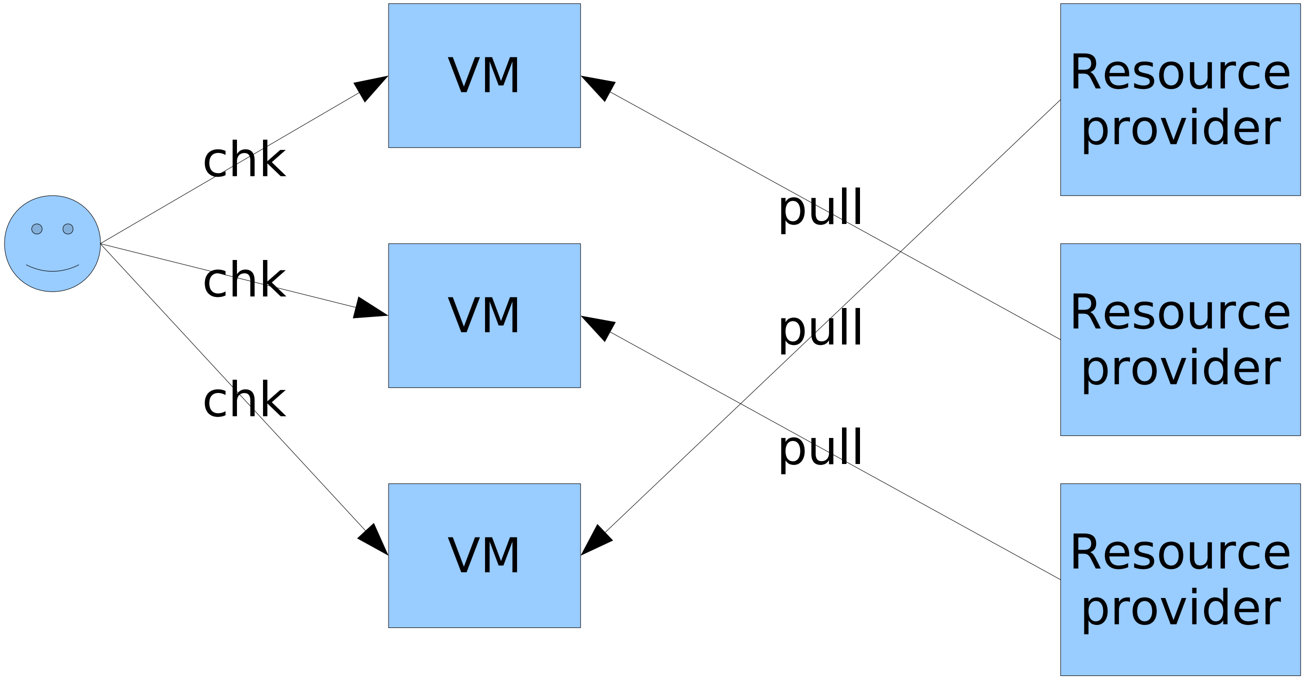
Process Migration with VMs

- VMs are a suitable vessel for migrating processes
  - VM images
    - Carry the execution environment with the job
  - Live-migration and checkpoints
    - Store and transfer execution state

New paradigms

- No guarantees necessary
  - Pay per successfully executed time segment
  - Resource providers can bid for VM state, knowing the extent of a time segment
- No software matchmaking
  - Run anywhere
  - Only hardware dependencies
- Redundancy
  - Empirically find more suitable resources
  - Pay only for fastest executed time segment

Example: Cycle scavenging / distributed computing



Considerations for live-migration

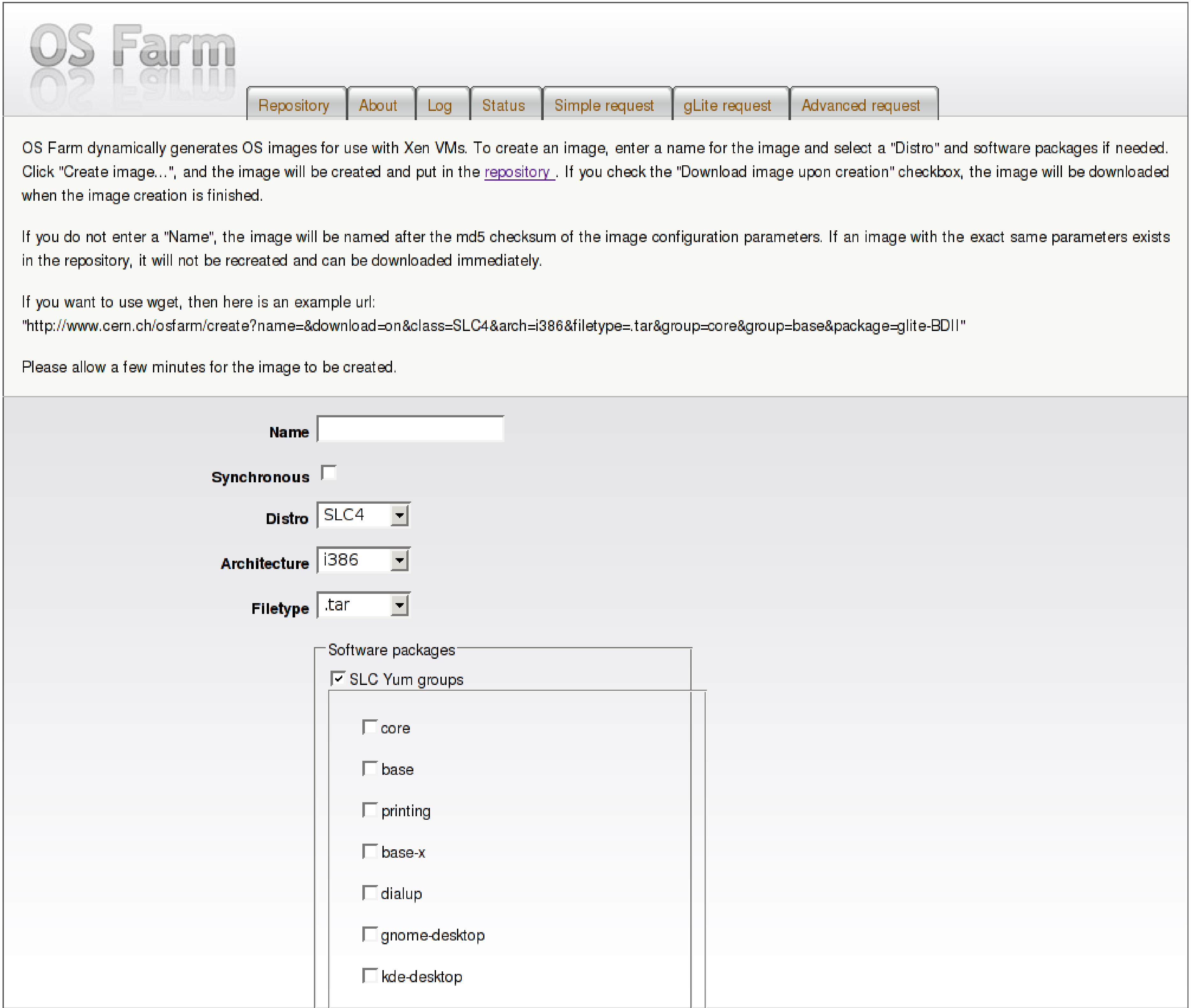
- Self-contained VMs
- Provide execution environments
  - OS Farm - virtual appliances
- Networking constraints
  - Must retain IP address
- Transferring VM images
  - Content-based addressing

Self-contained VMs

- Problem with traditional live-migration
  - Needs active receiver
  - Needs central storage server (NFS)
- Need to reduce dependencies in fabric
  - Example VM: ttyLinux
  - Root filesystem in RAM
  - Additional block devices attached dynamically ('xm block-attach')
  - 64 MB RAM = 64 MB VM

OS Farm

- On-demand generation and repository for VM images
- SLC3, SLC4 Xen VM images
    - User selectable yum groups and packages
- Virtual appliances
    - gLite services (glite-CE, WN, SE, etc.)
- x86, x86_64 architectures
- Different image formats
    - .img (raw), tar and gzipped tar archives
- http://cern.ch/osfarm



OS Farm dynamically generates OS images for use with Xen VMs. To create an image, enter a name for the image and select a "Distro" and software packages if needed. Click "Create image...", and the image will be created and put in the repository . If you check the "Download image upon creation" checkbox, the image will be downloaded when the image creation is finished.

If you do not enter a "Name", the image will be named after the md5 checksum of the image configuration parameters. If an image with the exact same parameters exists in the repository, it will not be recreated and can be downloaded immediately.

If you want to use wget, then here is an example url:
"http://www.cern.ch/osfarm/create?name=&download=on&class=SLC4&arch=i386&filetype=.tar&group=core&group=base&package=glite-BDII"

Please allow a few minutes for the image to be created.

Name

Synchronous □

Distro SLC4

Architecture i386

Filetype .tar

Software packages

☑ SLC Yum groups

□ core

□ base

□ printing

□ base-x

□ dialup

□ gnome-desktop

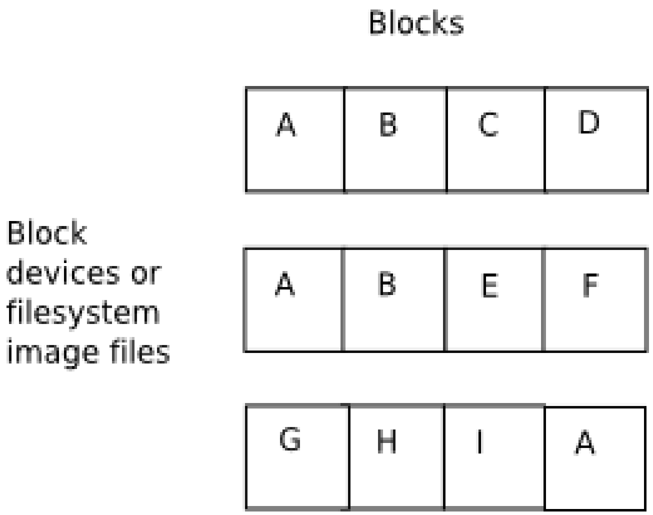
□ kde-desktop

Networking constraints

- Must retain IP address
- Solutions
  - VPN – being tested
    - Has limited scalability
    - Allows persistent connection
  - Globally unique private IP addresses
    - > 16 M of them
    - Depends on IP masquerading
    - Can allow persistent connection through gateway
  - One-way connection establishment
    - No need to retain IP address unless applications depend on it

Content-Based Addressing

- Speed up transfer of VM images over the network
- Block contents are calculated with hash algorithm



Blocks

Block devices or filesystem image files

A B C D

A B E F

G H I A

Content-Based Addressing

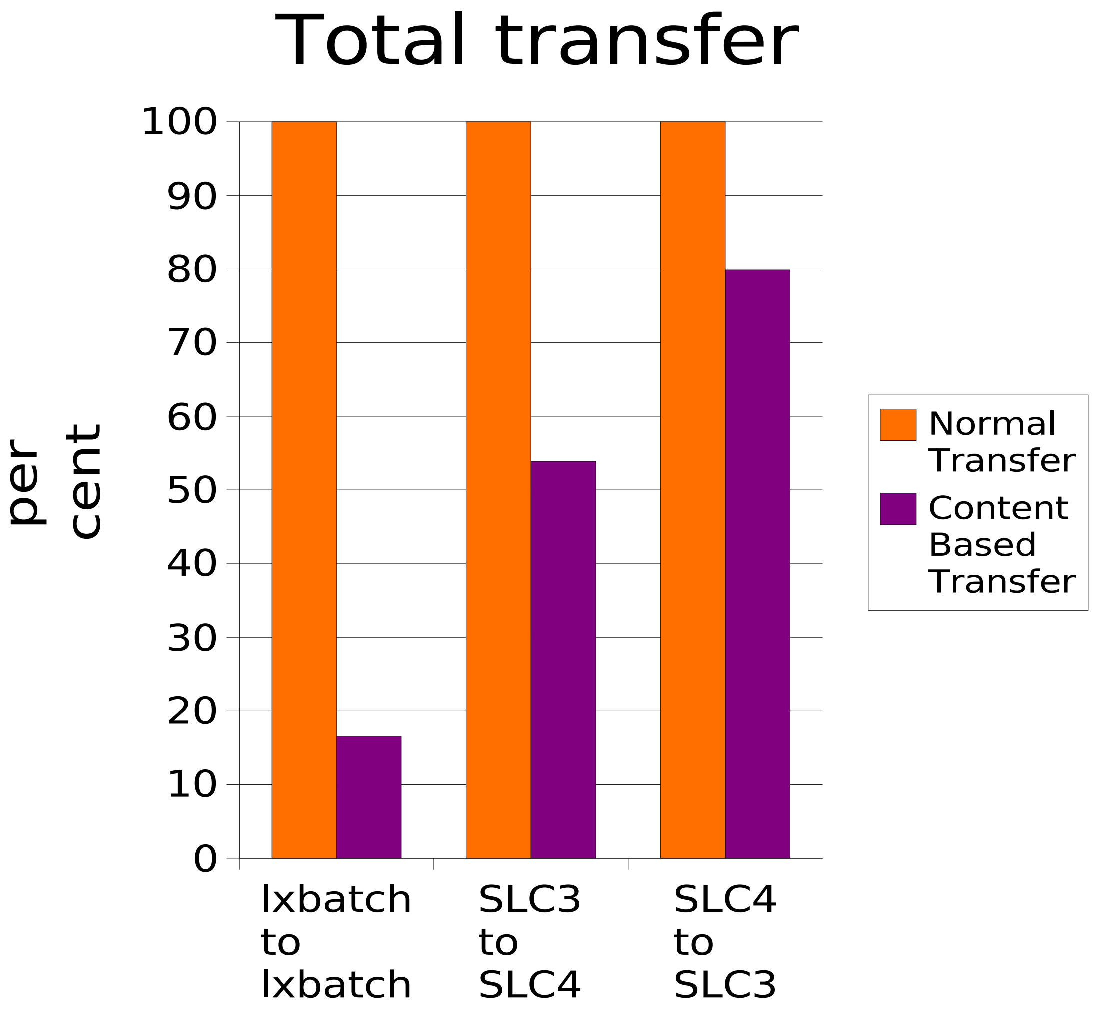
- Ext2 and Ext3 filesystems' files are 1k, 2k, or 4k aligned
- Common blocks are called “hot” blocks
- If hot blocks already exist on the target VMM machine, only cold blocks need to be transferred

Experimental Analysis

- Using SHA (Secure Hash Algorithm)
  - 20 bytes per block
- Two lxbatch root filesystems (5.3 GB)
  - 84 % hot blocks
- SLC3 (343 MB) and SLC4 (762 MB)
  - SLC3 -> SLC4
    - 48 % hot blocks
  - SLC4 -> SLC3
    - 22 % hot blocks

Estimated Data Transfer

- Hash table adds overhead
  - SHA: 0.48 to 2.0 %
  - MD5: 0.39 to 1.6 %



Questions?