Dynamic Extension of a Virtualized Cluster by using Cloud Resources
CHEP 2012
Thomas Hauth, Oliver Oberst, Günter Quast
Why Virtualization? Why Cloud?  

HEP Analysis Workflow

- HEP users tend to harvest all available resources for the final analysis steps  
  → Heterogeneous environments  
  → Strong confinements to the setup (SLC, VO software,....)

- Is there a possibility to use non-HEP resources (e.g. shared university clusters)?  
  → YES, if we use virtualization!
General Benefits of Virtualization

- Easier access to non-HEP resources
  - Enables use of private and public Clouds
  - Shared resources at universities (see ViBatch)
- Legacy support
  - Different OS/software versions
  - “CERN software appliance”: CERNVM
- Acceptable Performance overhead
  - Enables usage of additional resources!
- Temporary setup of environments
  - Test-deployment of infrastructures and development environments
  - Teaching, Schools (Gridka School, Statistics WS, …)

![Bar chart showing performance comparison between Nativ and Virt for Toy MC and CPU Benchmark (flop).]
Dynamic Virtualization of a Cluster

- **Isolated Computing Cluster**
  - Each group has sep. cluster
    - Administration overhead
    - Can not cover peak loads

- **Shared Computing Cluster**
  - All groups share one cluster
    - Setup compromise not always possible
    - Load-balancing by fair-share

- **Dynamic Partitioned Cluster**
  - Configure cluster in real-time with VMs
    - Allows any software/OS configuration
    - Virtualization layer hidden
    - Load-balancing by fair-share
Dynamic Virtualization of a Cluster
ViBatch – the Concept

User Interface

Batch server

Worker node

User Interface

Batch server

Virtual Worker Node

- Start VM
- Run job in VM (wait…)
- Stop VM

Worker Node

Fully transparent for the user!
Dynamic Virtualization of a Cluster
ViBatch – Implementation

- ViBatch
  - Wrapper script (Prologue/Epilogue) around the actual computing job inside the batch system
  - Virtualization system hidden from the users
  - Technical Details:
    - VM management via libvirt
    - Batch system: MAUI/Torque
    - Easy portable to other batch systems
    - Currently uses SLC 5 VM images with CERNVMFS installed (for use of CMS Software)

- In production usage for ~1.5 years now.

https://ekptrac.physik.uni-karlsruhe.de/trac/BatchVirt
Dynamic Virtualization of a Cluster
ViBatch – Deployment

- ViBatch ships with helper scripts to ease deployment of VMs
  - VM image deployment/configuration
    - VM template is copied to local storage of all selected Worker Nodes
    - Libvirt setup on the WN via automatic VM XML configuration creation
    - Central ViBatch configuration files

![Diagram of ViBatch deployment process]

**Diagram Explanation**

- **Admin**
  - SSH connection
  - `install.sh`
    - Iterate over all nodes
    - Execute further scripts
  - Execute `setupClient.sh`
  - Tailored scripts for each worker node
    - `templateDeploy.sh`
    - Further execution

**Key Points**
- Helper scripts simplify deployment process.
- VM images are efficiently managed and distributed.
- Libvirt setup streamlined with automatic configuration.
- Centralized configuration files ensure consistency.

---

*Intitut für Experimentelle Kernphysik*
Dynamic Virtualization of a Cluster

ViBatch – Cluster Storage

- Cluster FS has to be passed through to VM (user directories)
- No native Infiniband support within the Vms → no (direct) Lustre Mount possible
- Solution: export Lustre mounts via NFS per host
  - Scalablility: All HW nodes export their own Lustre mount to their Vms
  - Automatic /etc/exports creation per host
Dynamic Virtualization of a Cluster
ViBatch@KIT II – Virtual Infrastructure

User Portal 1 (SLES 11)
4 Cores / 16GB RAM

VM User Portal (SLC5)
4 Cores / 16 GB RAM

Worker Nodes
native queues

Virtual Worker Nodes
Virtual queues

VM Infrastructure Portal

Maui/Torque Server
CVMFS Proxy
Admin Server (logserver, dhcp...)

T3 Storage Element
T3 siteBDII
OpenVPN server
Dynamic Virtualization of a Cluster
ViBatch – Production environment

- Well accepted by users:
  - Private MC productions
  - Toy MC and Simulated CMS events
  - General data/MC analysis

- Experiences:
  - Generally stable:
    - ViBatch → stable
    - Instabilities are related to the cluster FS (lustre)
    - ViBatch copes with those problems automatically

![Graph showing failed, succeeded, and total jobs over time]

0 500 1000 1500 2000 2500
04.02 08.02 13.02 18.02 22.02 27.02 03.03

failed succeed succeed

Red: failed
Yellow: succeed
Green: succeed
Going to the Cloud

- **Typical scenario** during HEP analyses: During **periods of high demand** (before conferences, finishing publications, etc.) local clusters are often completely occupied.
  - Buying a new Cluster is expensive and most of the time it will be underutilized.
  - **Solution**: **Cloud burst**. Elasticity of Cloud Providers allows to dynamically add / remove Cloud resources and thus to handle peak loads.
Batch Resource extension in the Cloud ROCED

- Extension of local batch server infrastructures to Cloud resources
  - Dynamically, depending on the occupancy of the local infrastructure
  - Cost calculation, able to choose from different Cloud providers based on current cost
  - Completely transparent to the user

Implementation: ROCED
- Compatible with
  - Batch-sys: MAUI/Torque, SGE
  - Cloud API: EC2/Eucalyptus, OpenNebula
Batch Resource extension in the Cloud
ROCED – Detailed Workflow

1. Booting
2. Integrating
3. Working
4. Disintegrating
5. Down
6. Disintegrated
7. Site Adapter
8. Integration Adapter

**ROCED Broker**
decides how many machines have to be started or shut down

Monitor PBS queue
Integrate with PBS
Boot needed machines

22/05/12
Oliver Oberst
Intitut für Experimentelle Kernphysik
Batch Resource extension in the Cloud

**ROCED@KIT**

- ROCED runs on the same machine as the local batch server.
- Local batch system communicates with its nodes and users via TCP.
- Commands to the OpenNebula host are sent via XMLRPC call.
- The Communication between the Cloud nodes, ROCED and the Cloud Server are done via SSH.
- **No modifications to the firewall (besides VPN tunnel) needed.**
Batch Resource extension in the Cloud
ROCED VPN in Detail

Virtual Machine

init boot

booting

create & copy
certificate

init certificate
creation

get VPN IP

integrating

connect
to LDAP, storage

revoke
certificate

working

execute job
Batch Resource extension in the Cloud

ROCED Latest Developments

- OpenNebula Adapter
  - In use at KIT SCC Cloud
- Integrated monitoring
  - In development: web based monitoring output
Current Development: Fusion of ViBatch+ROCED

- **Fusion of Dynamic Virtualization and Batch-to-Cloud Extension**
  - Extension of local dynamically virtualized resources within private and public Clouds (cloudburst)

- **Current Status:**
  - First test setup running.
    - Cloud nodes setup manually at remote KIT Cloud site
    - SLC5 worker nodes using CERNVMFS
    - Gathering performance/reliability statistics

- **To be implemented:**
  - Central ViBatch and ROCED VM deployment
    - Deployment framework has to be adopted to OpenNebula/Cloud contextualization support
    → Automatic cloud site preparation
Summary/Outlook

- Dynamic virtualization and extension of local clusters:
  - Enables the usage of non-HEP only resources
  - ViBatch in KIT IEKP production usage for ~1.5 year
  - ROCED+ViBatch transition into production usage ongoing

- Outlook:
  - Testing a combination of Grid and virtual Cloud resources (combine local glideinWMS Grid farm and our Cloud extended (ROCED) farm)
Questions?
BACKUP
Dynamic Virtualization of a Cluster
ViBatch – Workflow

User submits jobs to desired queue

Prologue Script executed at jobs startup
1. Create VM image from template
2. Start VM
3. Wait for VM to finish booting (check for lockfile)
4. Pipe job to VM (ssh)
5. Run job in VM

Worker Node
hypervisor with libvirt
API: Xen, KVM, ...

Native Host OS runs on Hardware

Guest OS 1 runs in VM 1
init script lockfile

VM templates

VM1 VM2 VM3
VM templates
local disk

Batch Server with different queues

Native Batch Queue

VM Queue 1 e.g. SLC

VM Queue 2 e.g. Debian

VM Queue 3 e.g. SuSE

cluster filesystem
Dynamic Virtualization of a Cluster
ViBatch@KIT

Hardware:
- Worker Nodes
  - 2x Intel Quadcore Xeon X5355 (8 cores, 16GB RAM)
  - Diskspace per node: 750GB
- Cluster FS
  - Lustre (~350TB, Infiniband)

Software
- Host OS
  - SuSE Enterprise 11sp1 (SLES10sp2) 64 Bit
  - Kernel 2.6.32.54-0.3-default
  - Virtualization: kvm-kmod-3.3-1, qemu-kvm-1.0-1, libvirt-0.9.11-1
- VM OS
  - SLC 5 64 Bit
- Batch System
  - MAUI/Torque
    - MAUI 3.3
    - Torque 2.5.1