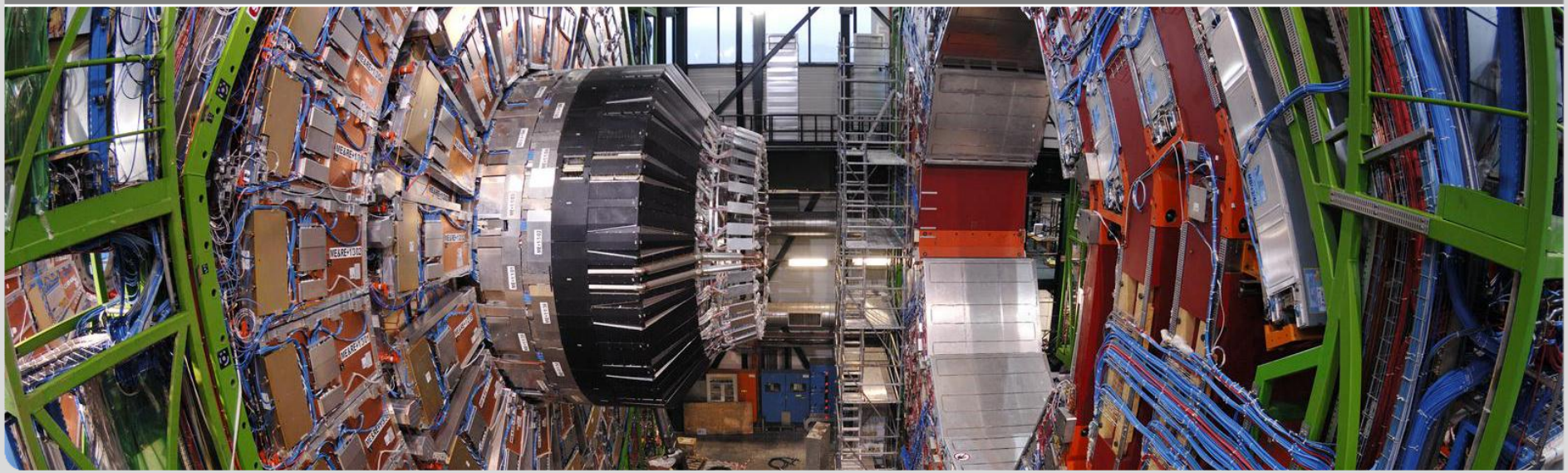
Virtualization & Cloud-Computing

A. Scheurer on behalf of the Karlsruhe and Freiburg virtualization Team

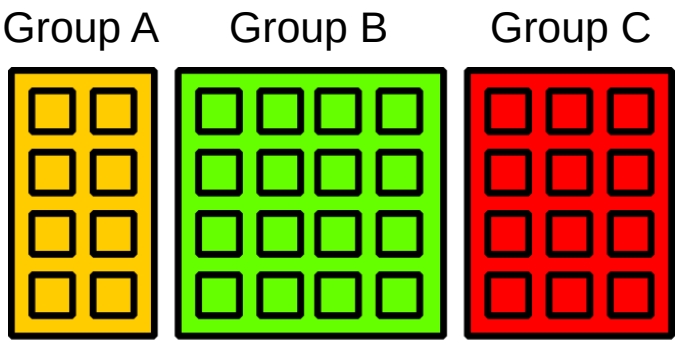
4th Annual Workshop of the Helmholtz Alliance "Physics at the Terascale“
Dresden, 01. – 03. December 2010

Institut für Experimentelle Kernphysik



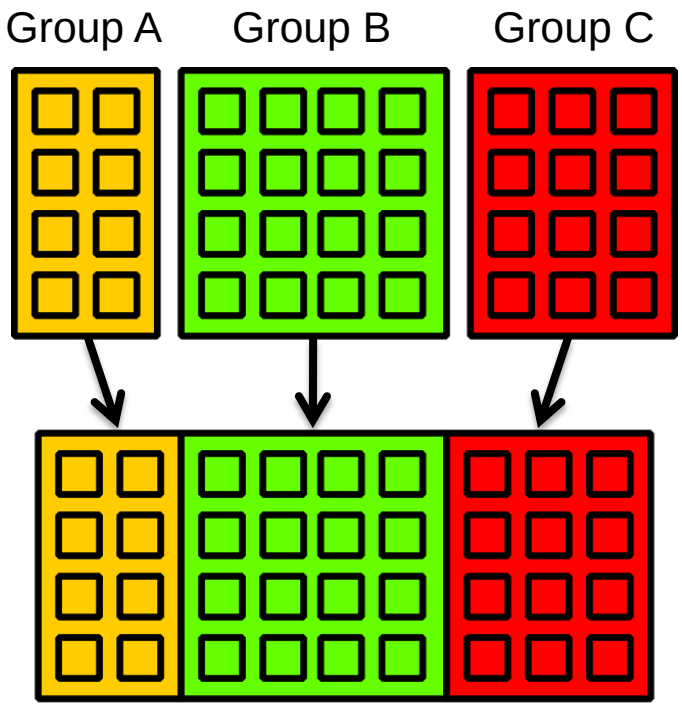
HPC Cluster Models

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



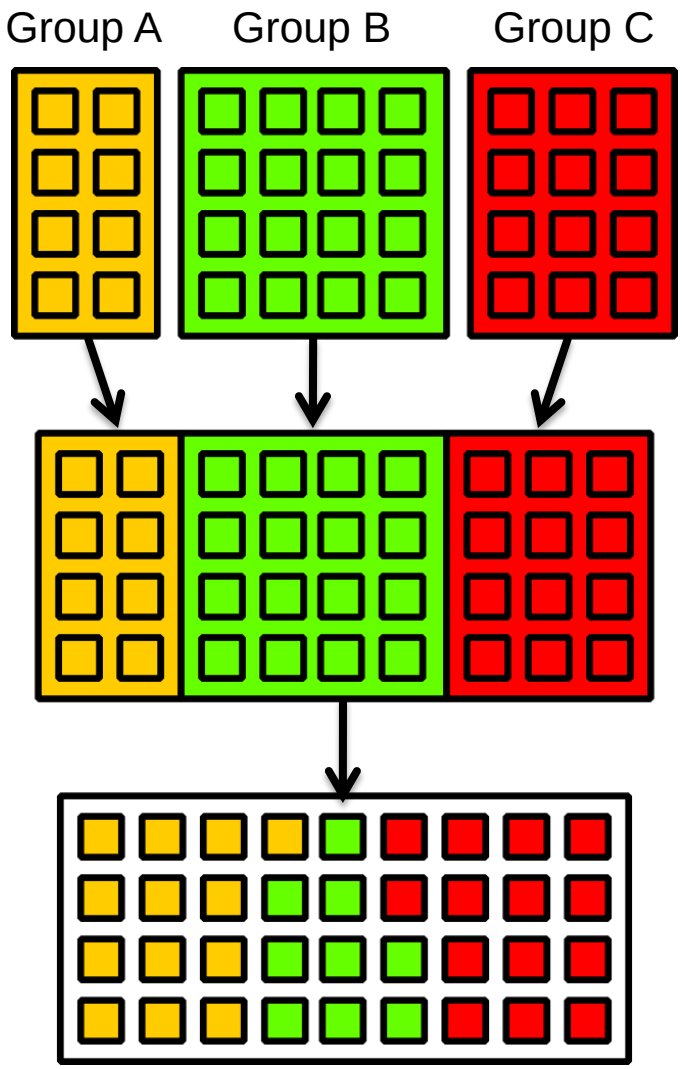
HPC Cluster Models

- Isolated Computing Cluster
  - Each group/institution 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



HPC Cluster Models

- Isolated Computing Cluster
  - Each group/institution 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

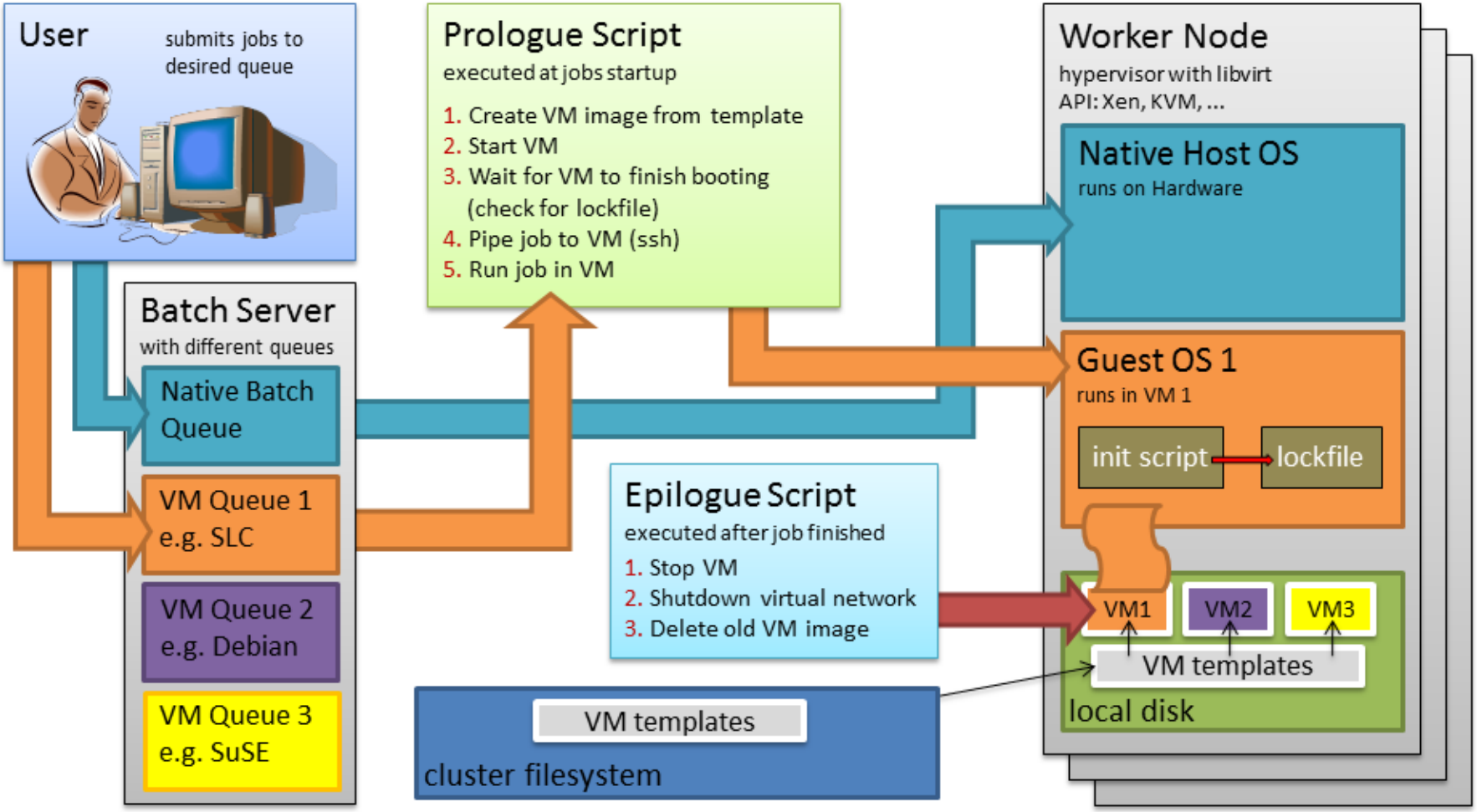


Interoperability of ViBatch

- ViBatch uses `libvirt`
- `libvirt` is a library providing an API to basically all common virtualizers
- Command line interface: `virsh`
- GUI: `virt-manager`
- Provides bindings to: C/C++, Python, Perl, Ruby, Java, C# and others
- Security aspects:
  - Encrypted (TLS, x509 cert) remote management
  - Authentication with Kerberos & SASL
  - Local access control, PolicyKit

`libvirt` provides everything needed to manage and control a dynamic virtual cluster → independent from the used hypervisor

The ViBatch Concept



Implementation: KIT IC1 Cluster

- IC1 Cluster
  - Shared between 9 institutions of KIT
  - Very different setup requirements, mostly parallel HPC
  - Without virtualization, no possibility to use the cluster (no SLC)
- IC1 Hardware & Software Setup
  - 1600 CPU cores
    - 200 x 8 core Intel Xeon X5355 (VT-x, 64bit)
  - 3200 GB memory
    - 2 GB memory / core
  - 400 TB Lustre filesystem
    - Exported via NFS to VMs
  - Operating system
    - Host: SUSE Linux Enterprise Server 11 (Kernel 2.6.27)
    - Guest: Scientific Linux CERN 5
    - Hypervisor KVM Version 2.6.36

Implementation: Freiburg University - I

Slide by Vahidenur Culha Flechl

- 140 nodes
- IBM-Bladeserver HS21XM
- Intel Xeon E5440 2.83 Ghz Processor
- 12 MB cache
- 1333 Mhz bus speed
- 16 GB RAM (2 GB per Prozesskern, 8 Prozesskerne)
- SL 5.0 64 bit

Virtualisierung im BWGrid

Unsere Motivation für Virtualisierung:

- Mehr Kapazität aus den bereits vorhandenen Ressourcen
- Niedrige Kosten (weniger Hardware, Platzbedarf, Stromverbrauch und Kühlung)
- Flexibilität (Migration, Snapshots ...)
- Hohe Verfügbarkeit
- Bessere Verwaltbarkeit und Sicherheit
- Verschiedene Betriebsysteme auf einer physikalischen Maschine

Implementation: Freiburg University - II

Slide by Vahidenur Culha Flechl

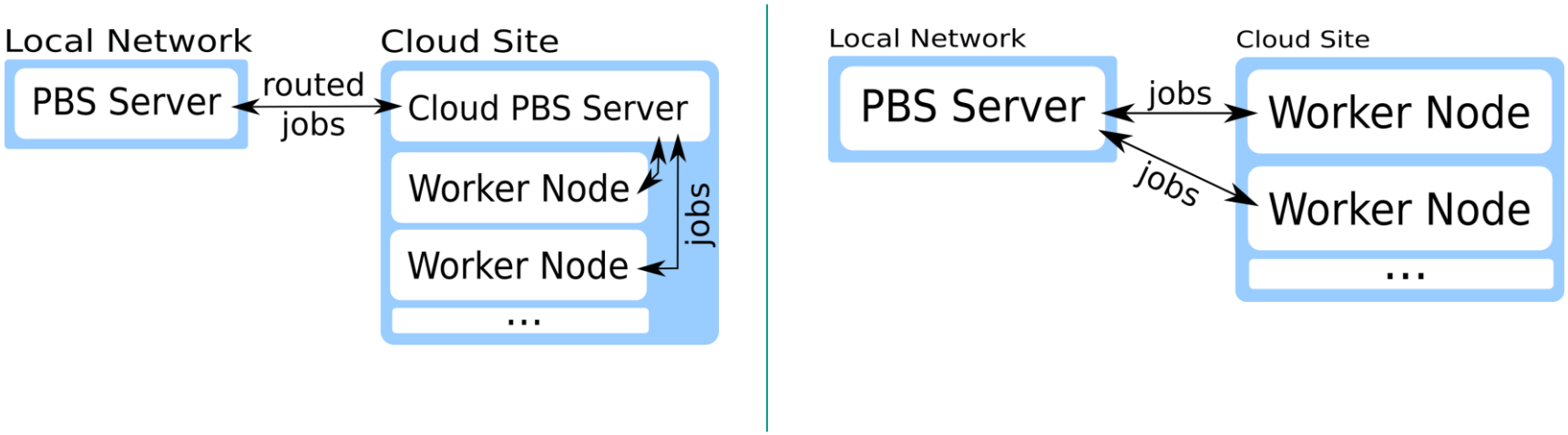
- **KVM:** Hypervisor
  - Wird von RedHat unterstützt
  - Einfach zu installieren und zu managen
- **LIBVIRT:** Backend fürs managen von virtuellen Maschinen
- **VIRSH:** Frontend fürs managen von virtuellen Maschinen
- **ViBatch*:** Dynamische Erzeugung von virtuellen Maschinen

*: Entwickelt von KIT

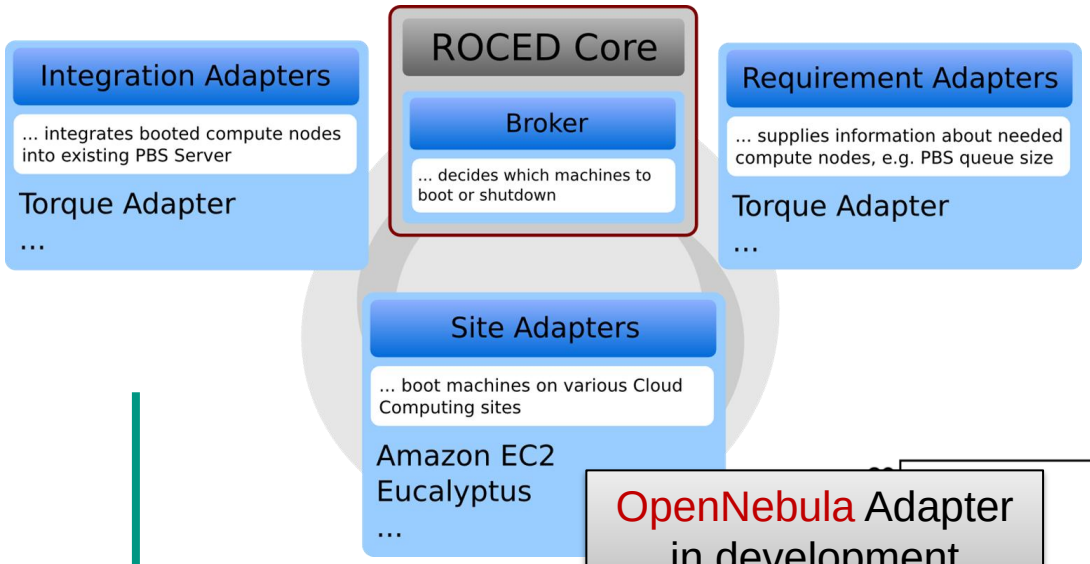
Further Virtualization Projects at KIT: Batch-to-Cloud Extension

- 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

- Two possible topologies tested and evaluated:



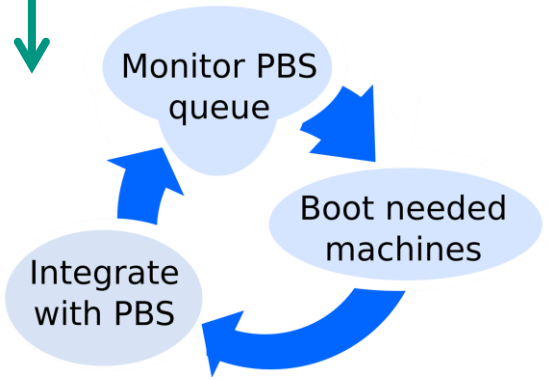
The ROCED Scheduler

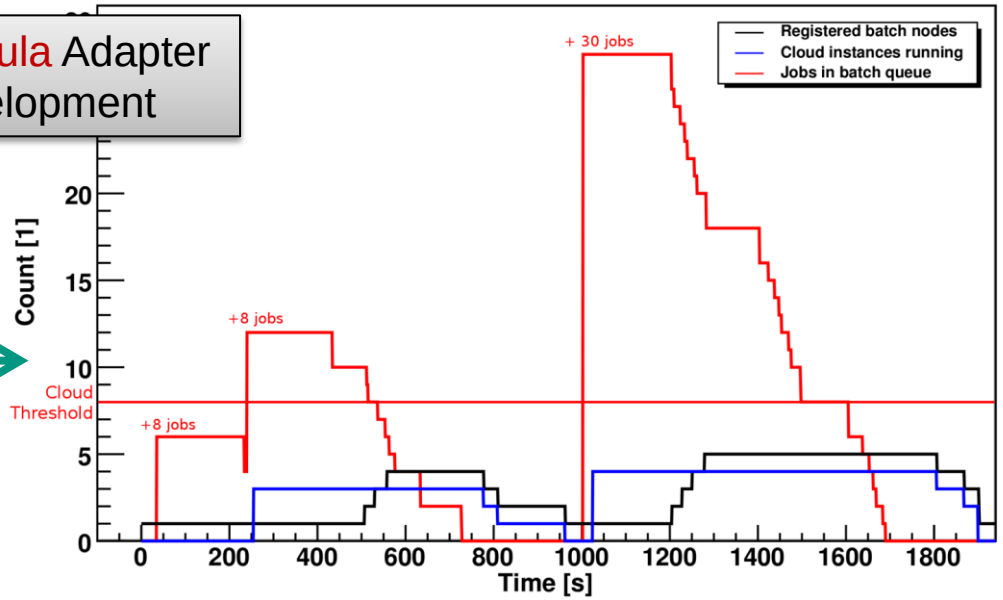


Responsive On-demand Cloud-Enabled Deployment

The **ROCED**-Meta-Scheduler

(Modular framework, written in Python)





Summary

- Virtualization offers the possibility to benefit from shared clusters
  - Flexibility
  - Easy administration
  - Security
  - Load-balancing
  - Performance loss small compared to benefit
- ViBatch Dynamic Cluster Concept
  - Easy to integrate
  - Light-weight & transparent
  - No batch system modification necessary, only pro/epilogue functionality
  - Independent from hypervisor
- ROCED Cloud Scheduler
  - Extension of local resources on demand
  - Still unsolved topic: HEP demand on storage (public vs. private Clouds)

The ViBatch Project

https://ekptrac.physik.uni-karlsruhe.de/trac/BatchVirt