TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Running Virtual Machines in a Slurm Batch System

Slurm User Group Meeting

September 2015

Ulf Markwardt
+49 351 - 463 33640
ulf.markwardt@tu-dresden.de

ZIH
Center for Information Services & High Performance Computing

Acknowledgements

These colleagues contribute to our VM / Slurm infrastructure:

- Matthias Jurenz - Slurm plugins/provisioning
- Danny Rotscher - Slurm scripts/programming
- Ralph Müller-Pfefferkorn - integrating the G-Lite infrastructre,
- Rene Jäkel - Open Nebula
- **Bert Wesarg** - programming of the VM scheduler,

Motivation

We have a new nice HPC system (#66 in Top 500):

- over 1 PFLOPS performance
- about 5 PB scratch file system
- heterogeneity
  - CPU (Westmere, Sandybridge, Haswell),
  - 9 different sitzes of memory (from 2 GB/core to 2 TB/node),
  - different accelerators
- Slurm is running fine with version 14.11.7 -
  (thanks to SchedMD for a few bug fixes...)

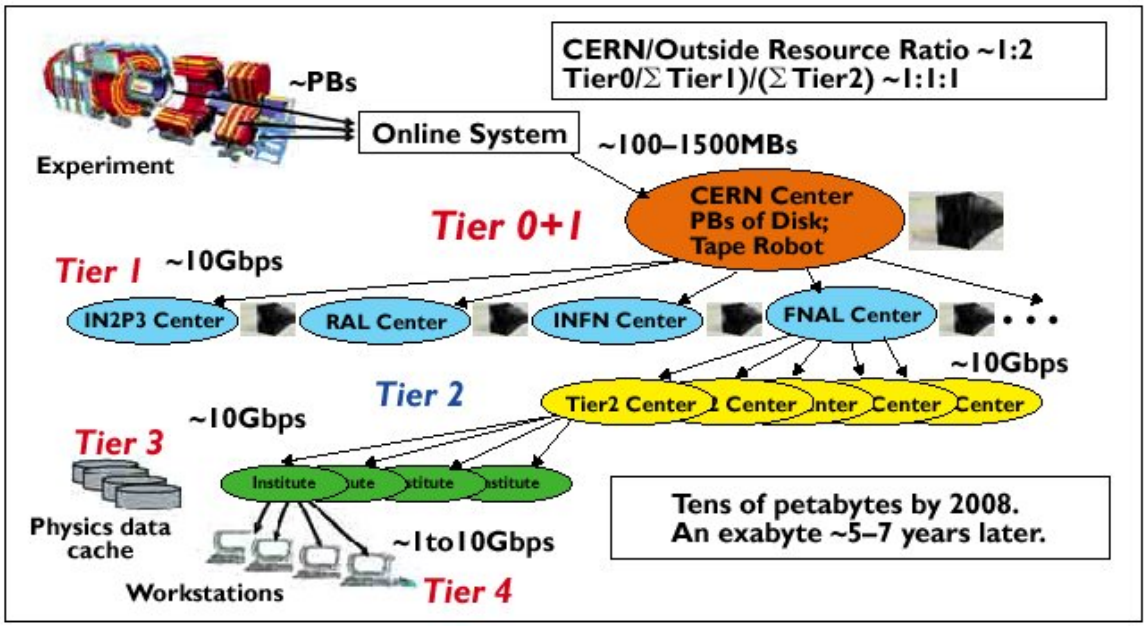
Motivation

We have a new nice HPC system (#66 in Top 500):

- over 1 PFLOPS performance
- about 5 PB scratch file system
- heterogeneity
  - CPU (Westmere, Sandybridge, Haswell),
  - 9 different sitzes of memory (from 2 GB/core to 2 TB/node),
  - different accelerators
- Slurm is running fine with version 14.11.7 -
  (thanks to SchedMD for a few bug fixes...)

But some research communities need a special access to use the resources easily...

Motivation



Source: CERN

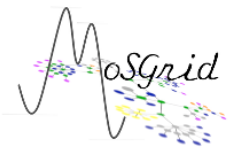
Requirements of scientific experiments providing data for thousands of physicists (eg. LHC @ CERN): →Grid computing

- scalable data infrastructure,
- well-defined software stack to enable reproducible analyses,
- easy-to-use, standardized access to compute resources

Motivation

Infrastructure works (too) well...

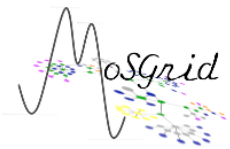
- Science gateways for other communities
  provide easy access to specific software and workflows
- rising demands for compute resources



Motivation

Infrastructure works (too) well...

- Science gateways for other communities
  provide easy access to specific software and workflows
- rising demands for compute resources



But HPC software is somewhat restricted
when it comes to specific versions:

- specific hardware (IB cards, accelerators),
- software dependencies (Lustre),
- vendor support policies.

Motivation

Solution: use virtual machines in an HPC / throughput system.

- arbitrary software stack (...),
- complete system isolation,
- fast start-up and shut-down
- stable and usable virtualization environment (KVM)

Requirements:

- hide virtualization from user
- multiple users can share a VM / image
- multiple images

Design

Testcase: How can we run jobs from ATLAS Grid on our HPC system?
Requirements:

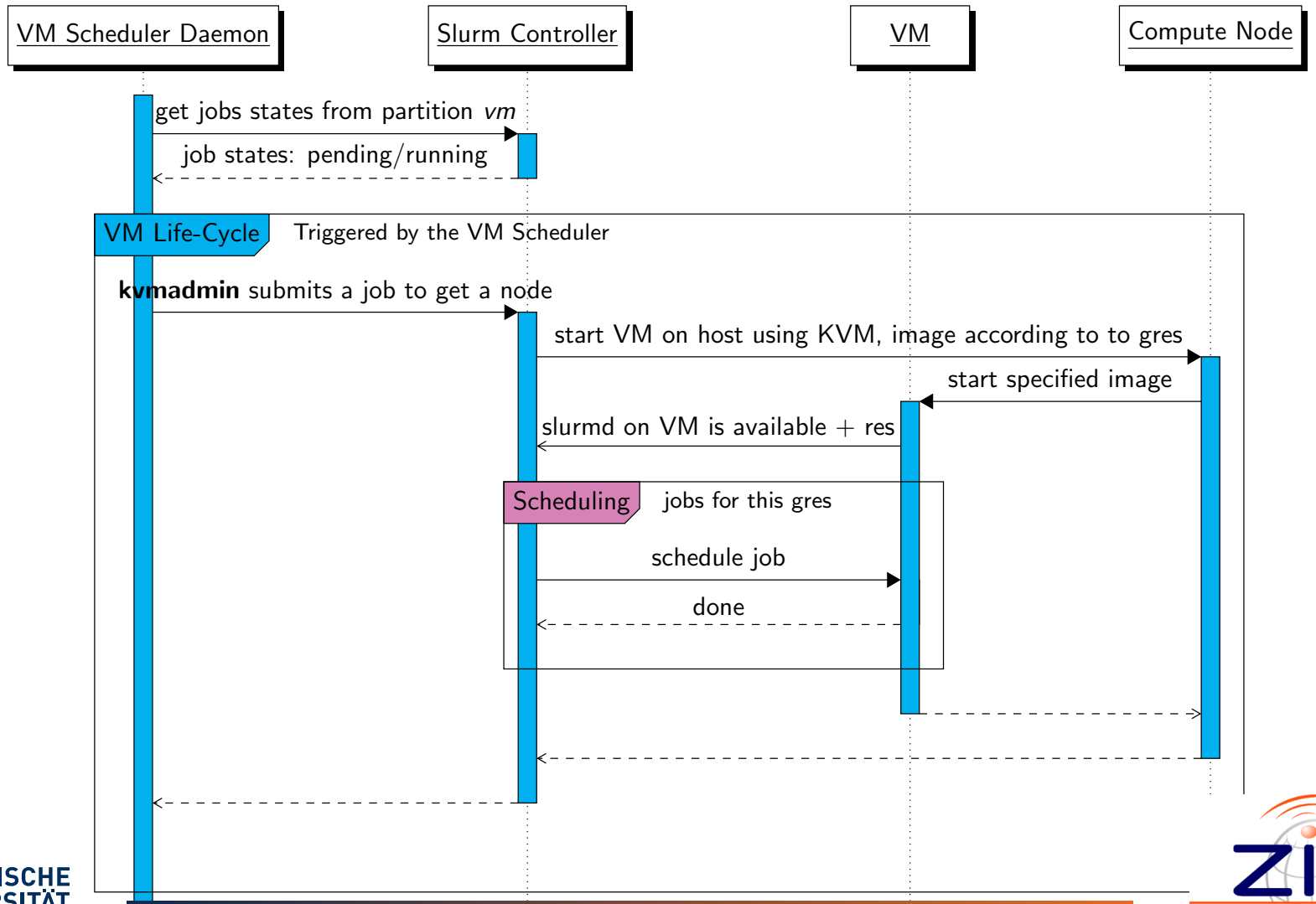
- Scientific Linux 6.6
- no persistent file systems
- about 40 GB local disk space needed

Requirements

- Slurm needs a unique hostname for each VM.
  - Start a VM with a well-defined MAC address,
  - Dynamically enter MAC/IP/name into local DHCP and DNS servers.
  - Once the VM powers up it then has exactly the hostname Slurm knows, like `vm-SL6-003`
- Jobs in the queue need *different* images
  - We use `--gres=vm_<imagename>`; node definition like: `NodeName=vm-SL6-[001-010] Gres=vm_SL6:24 ...`
  - We have plenty of (mostly unused) hostnames in the configuration, for each image.
- We have a maximum runtime of 7 days
- Garbage collector..

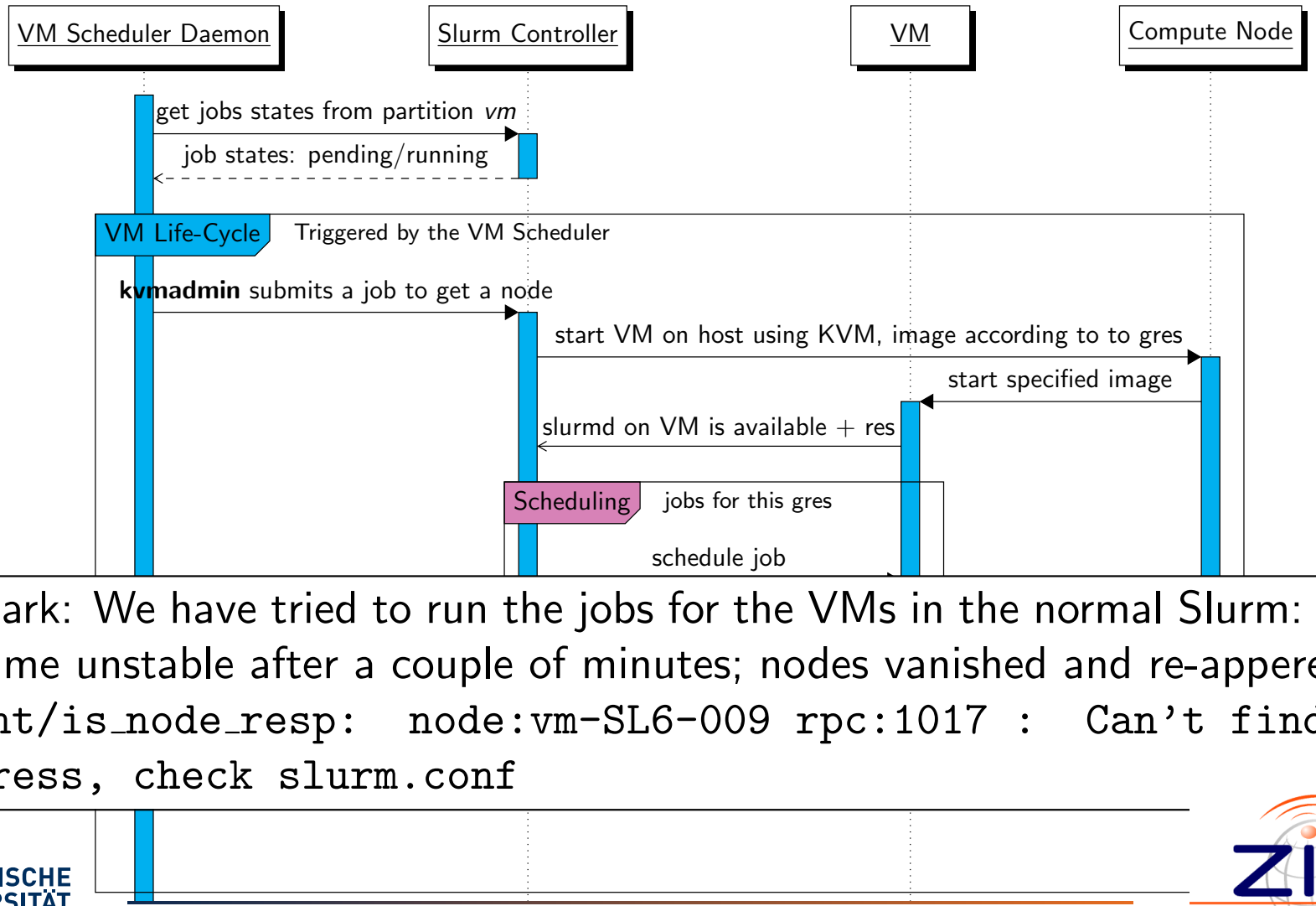
Process overview

A VM scheduler keeps track on the Slurm queue and on the VMs.



Process overview

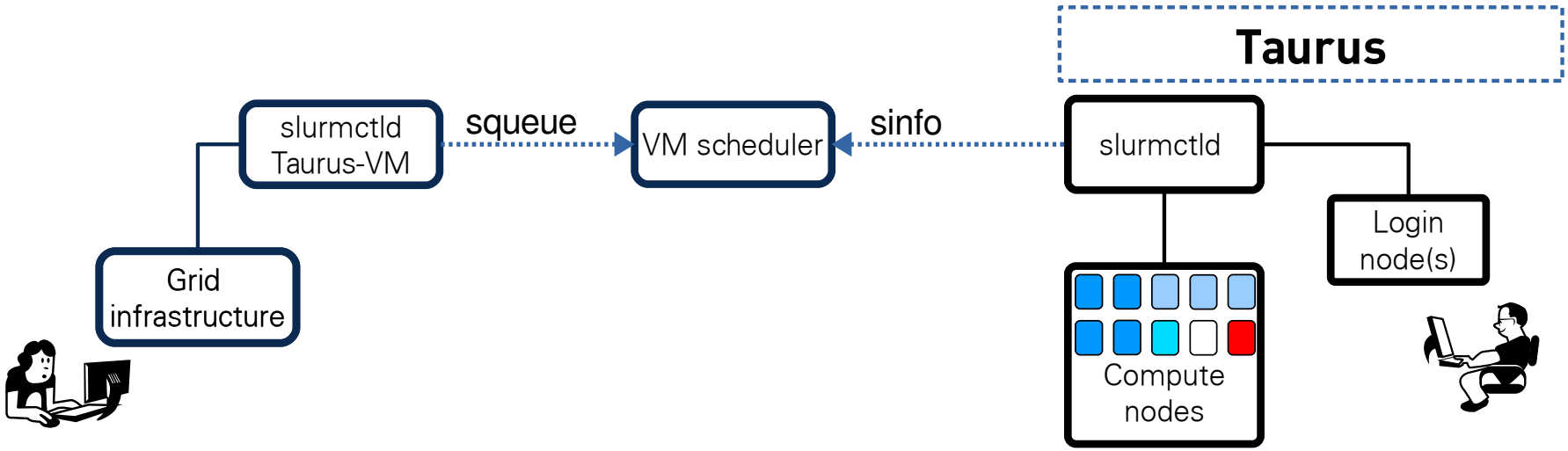
A VM scheduler keeps track on the Slurm queue and on the VMs.



Remark: We have tried to run the jobs for the VMs in the normal Slurm: This became unstable after a couple of minutes; nodes vanished and re-appered.

```
agent/is_node_resp: node:vm-SL6-009 rpc:1017 : Can't find an address, check slurm.conf
```

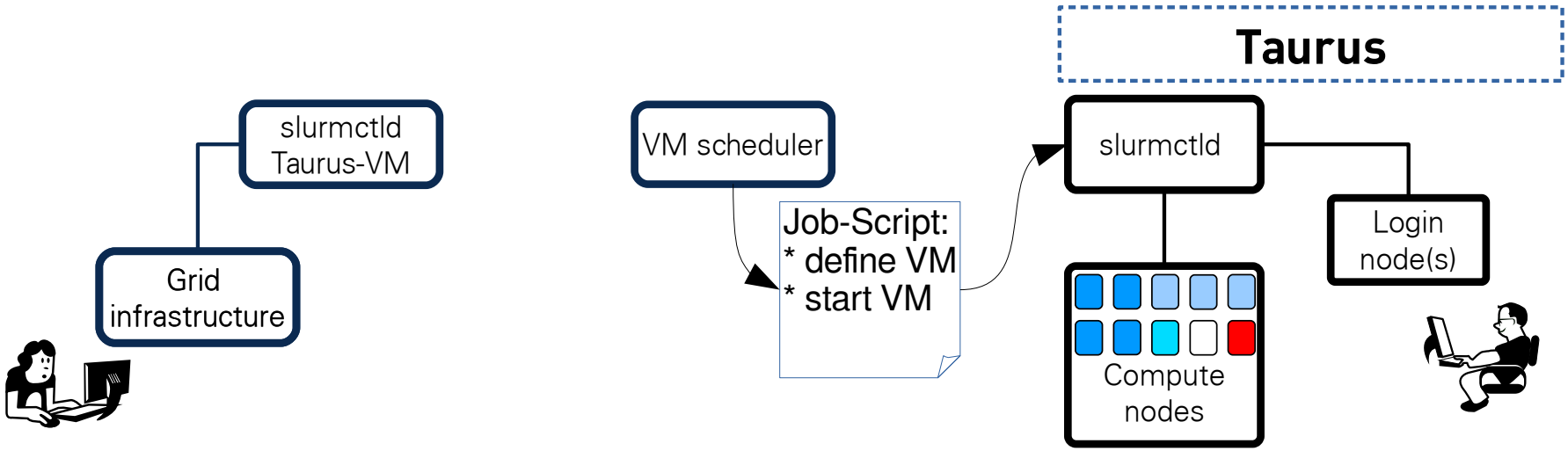
Design



Solution: Run the virtual machines in their own batch system.

VM scheduler looks at the VM queue and the hosts' Slurm.

Design

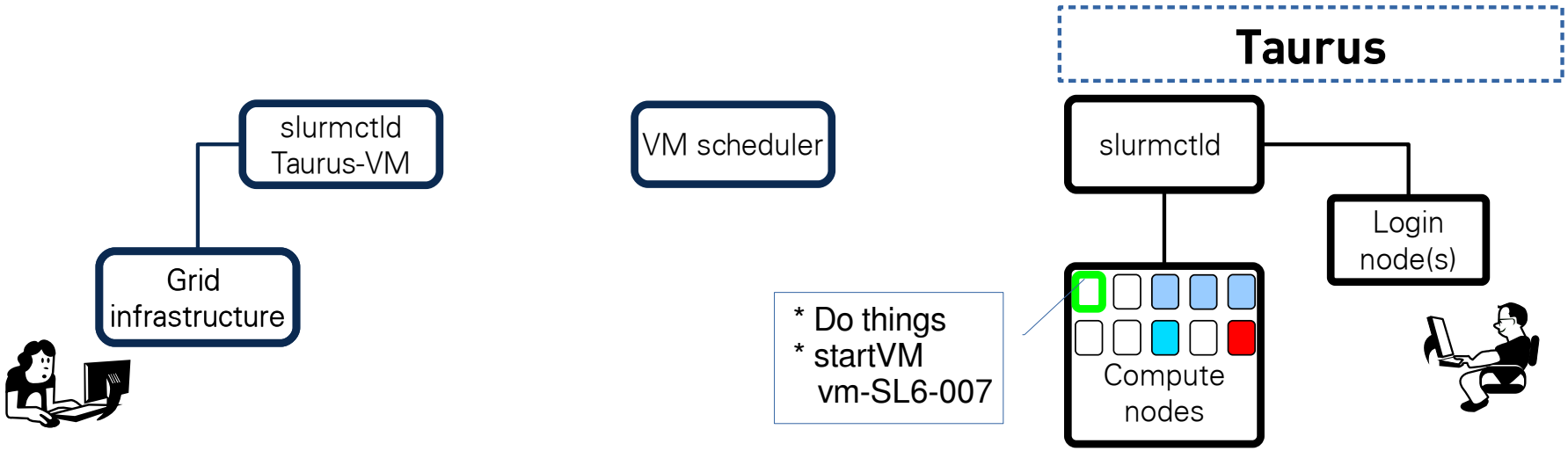


According to a set of rules, VM scheduler now schedules an exclusive job in the normal batch system to start a new VM with these attributes:

- name of the image
- MAC address
- hostname

The VM scheduler adds the new attributes to the DHCP and DNS servers.

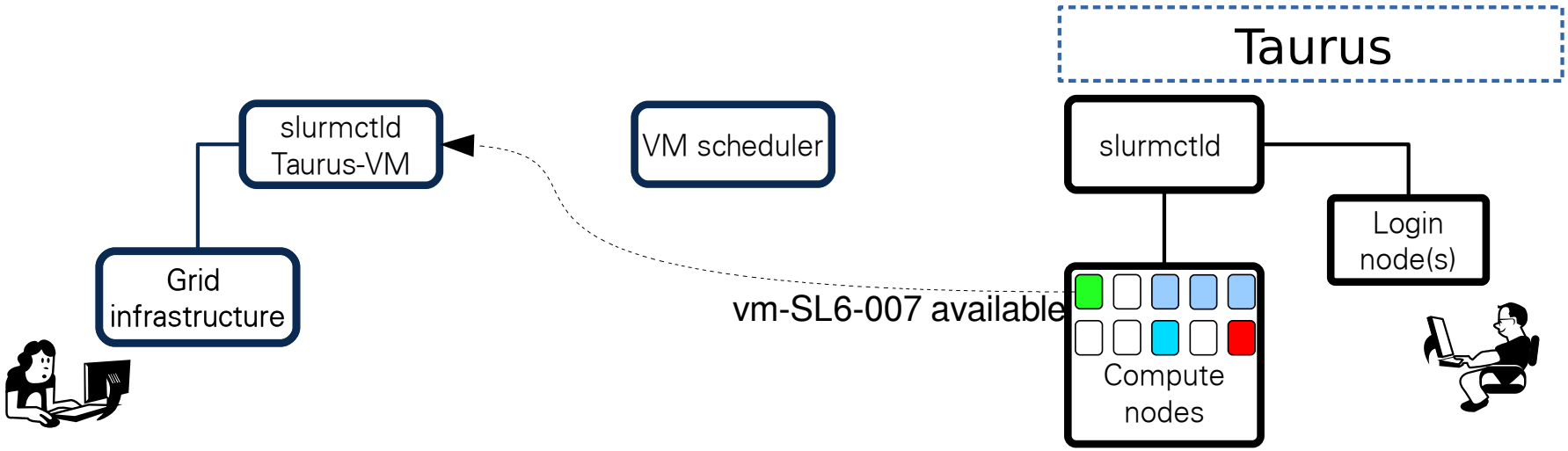
Design



Things to do when the job starts:

- create a raw file for the VM's /tmp
- create Slurm reservation for the VM in 7 days (maximum job run time)
- start the VM

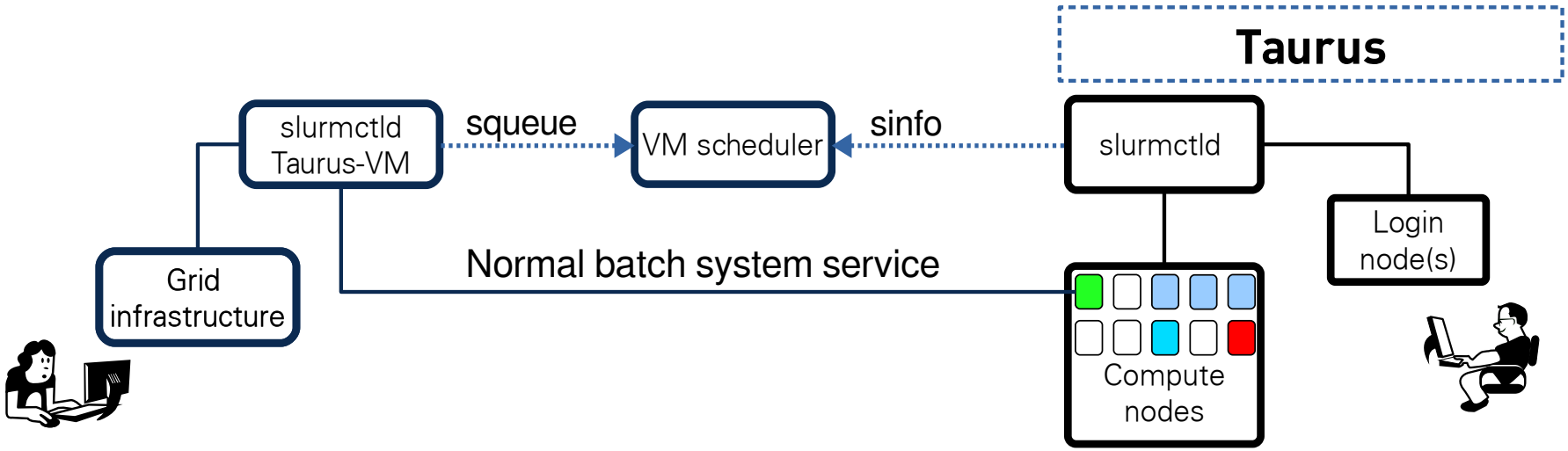
Design



At boot time:

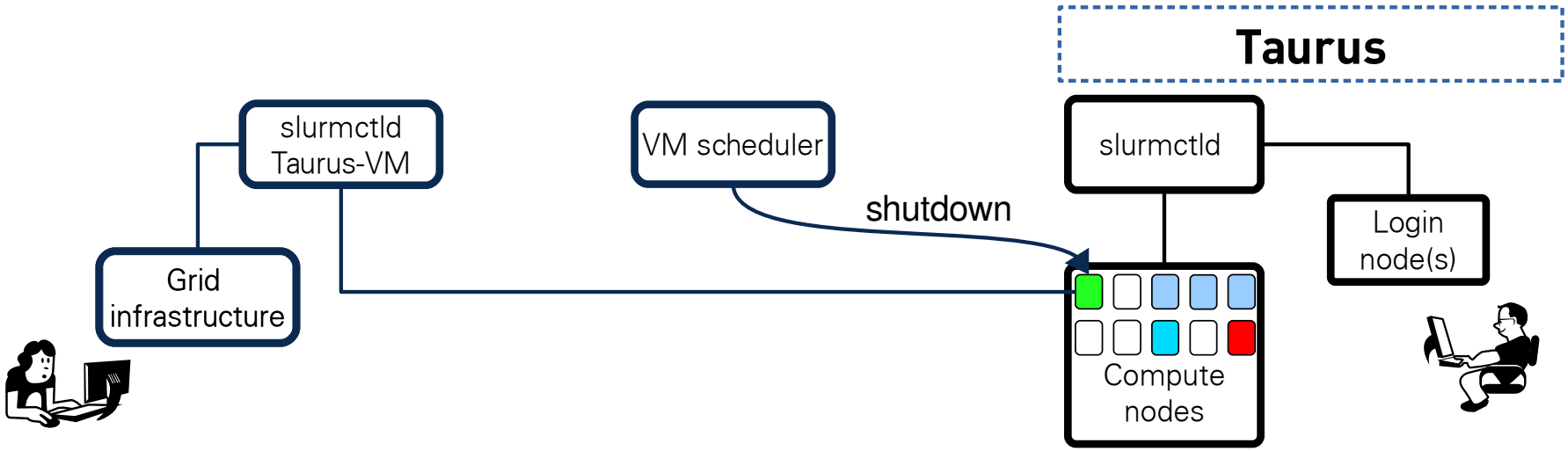
- get/expand a tarball including start scripts, Slurm binaries
- get `slurm.conf` and others
- `scontrol update state=idle ...`

Design



VM scheduler supervises the VM queue and the VM infrastructure...

Design



According to a set of rules, it now shuts down the VM:

- `scontrol update state=down ...`
- delete the entries in DHCP and DNS
- completely delete the virtual machine on the host
- clean up the temp space on the host

Provisioning

Our Slurm version and our config files change faster than images :-)

→ use the same Slurm binaries as the host

→ provision the current configuration to `/etc/slurm` at boot time

We already have a good provisioning framework...

Provisioning

Heterogeneous systems need robust provisioning:

- GPU / non-GPU
- energy accounting: IPMI-raw(Bull) / IPMI
- batch / interactive
- test nodes (hardware/software)

Provisioning

Heterogeneous systems need robust provisioning:

- GPU / non-GPU
- energy accounting: IPMI-raw(Bull) / IPMI
- batch / interactive
- test nodes (hardware/software)

Different files:

- slurm.conf (AcctGatherEnergyType)
- gres.conf
- prolog, epilogs, plugstack.conf (e.g. CPU/GPU frequency)
- sanity checks

Provisioning by filename

At the end of `slurm.conf` we `include slurm.local.conf` coming in different flavours (covering AcctGatherEnergyType):

```
slurm.local.conf.0.n_#ADMIN_NODES
slurm.local.conf.0.n_#COMPUTE_NODES
slurm.local.conf.1.n_#GPU_NODES
slurm.local.conf.remain
```

- Aliases are in `nodeset` syntax (`GPU_NODES=taurusi[2045-2108]`)
- Each host uses the files with a match of hostname/alias.
- Additional integer makes overloading possible for special cases.

Provisioning by filename

At the end of `slurm.conf` we `include slurm.local.conf` coming in different flavours (covering AcctGatherEnergyType):

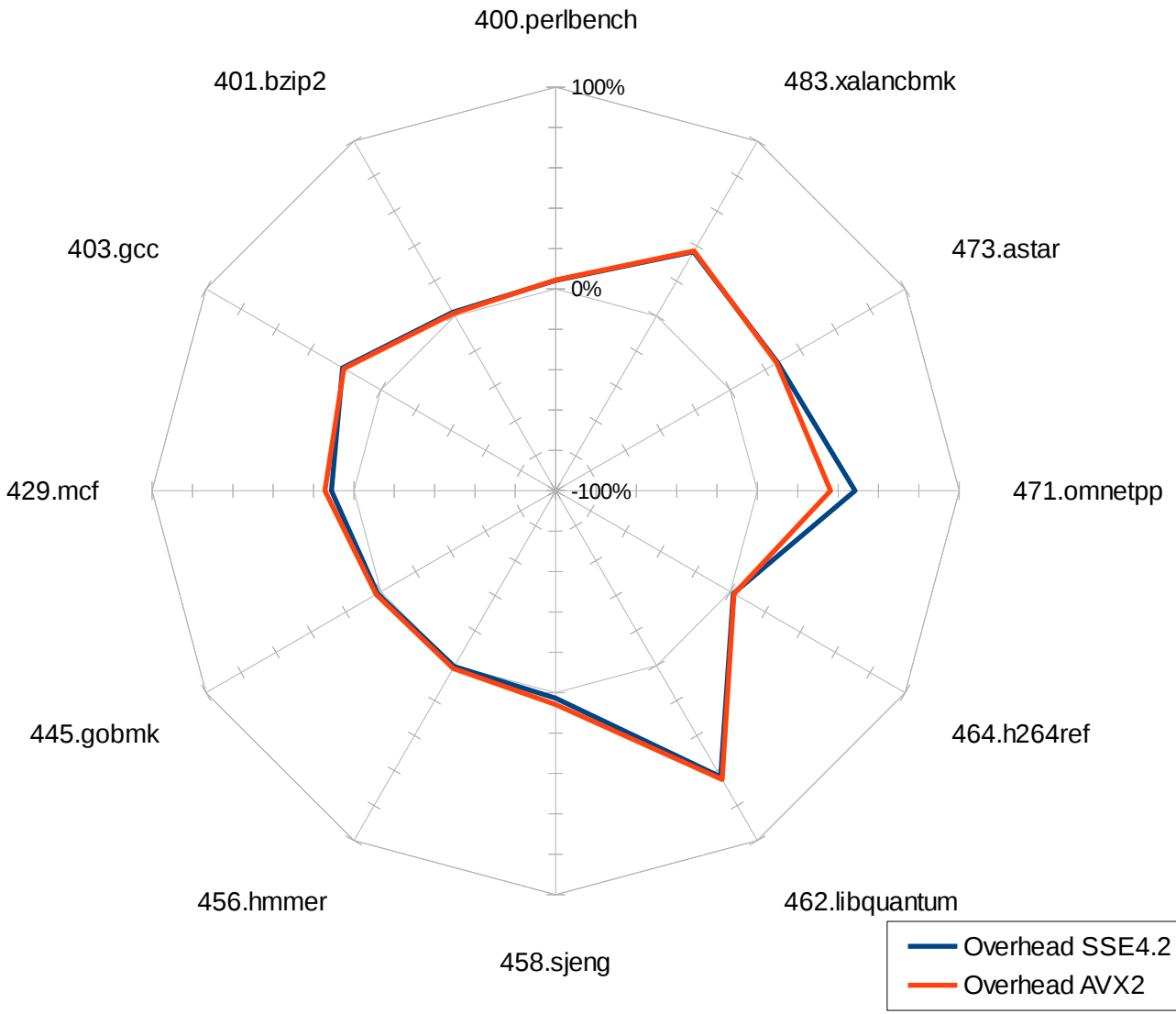
```
slurm.local.conf.0.n_#ADMIN_NODES
slurm.local.conf.0.n_#COMPUTE_NODES
slurm.local.conf.1.n_#GPU_NODES
slurm.local.conf.remain
```

- Aliases are in `nodeset` syntax (`GPU_NODES=taurusi[2045-2108]`)
- Each host uses the files with a match of hostname/alias.
- Additional integer makes overloading possible for special cases.

We have our configuration on a backed-up file system. Two ways of provisioning:

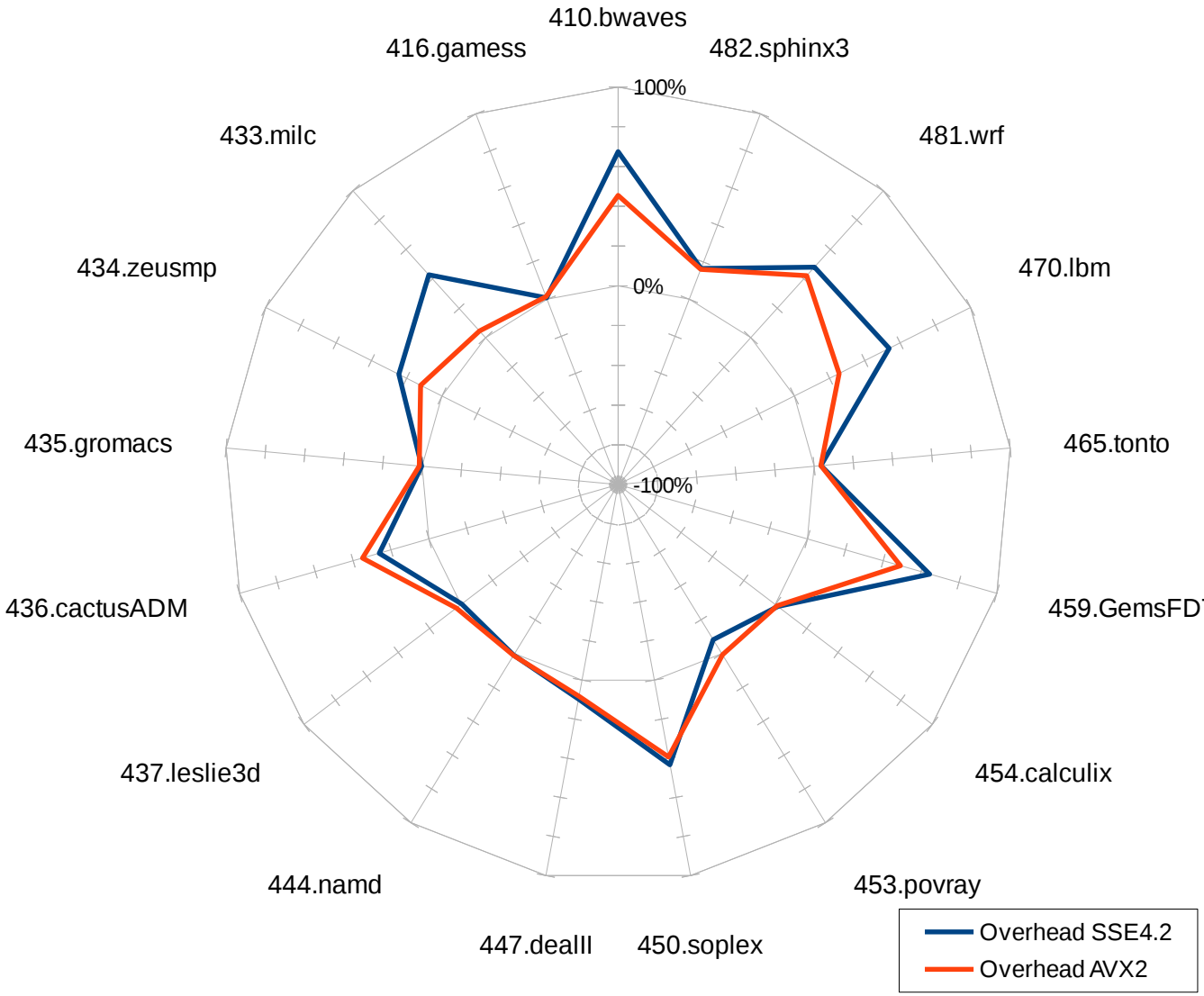
- Nodes with shared file systems (compute, login) *pull* their config during `/etc/init.d/slurm`.
- We *push* the configs for all other (admins).

Comparison between VM and Host with SPEC CPU



CINT2006	
400.perlbench	Programming Language
401.bzip2	Compression
403.gcc	C Compiler
429.mcf	Combinatorial Optimizatio
445.gobmk	Artificial Intelligence: Go
456.hmmer	Search Gene Sequence
458.sjeng	Artificial Intelligence: ches
462.libquantum	Physics / Quantum Compu
464.h264ref	Video Compression
471.omnetpp	Discrete Event Simulation
473.astar	Path-finding Algorithms
483.xalancbmk	XML Processing

Comparison between VM and Host with SPEC CPU



CFP2006	
410.bwaves	Fluid Dynamics
416.gamess	Quantum Chemistry.
433.milc	Physics / Quantum Chromodynamics
434.zeusmp	Physics / CFD
435.gromacs	Biochemistry / Molecular Dynamics
436.cactusADM	Physics / General Relativity
437.leslie3d	Fluid Dynamics
444.namd	Biology / Molecular Dynamics
447.dealII	Finite Element Analysis
450.soplex	Linear Programming, Optimization
453.povray	Image Ray-tracing
454.calculix	Structural Mechanics
459.GemsFDTD	Computational Electromagnetics
465.tonto	Quantum Chemistry
470.lbm	Fluid Dynamics
481.wrf	Weather
482.sphinx3	Speech recognition

Next steps

What about Linux containers / Docker?

KVM	**Container**
some performance degradation	same performance as native
moderate memory ovehead	very low memory overhead
sw completely independent from host	sw stack bases on kernel of the host
easy standalone configuration	modular configuration possible
migration and long runs possible	(like host)

Next steps

What about Linux containers / Docker?

KVM	**Container**
some performance degradation	same performance as native
moderate memory ovehead	very low memory overhead
sw completely independent from host	sw stack bases on kernel of the host
easy standalone configuration	modular configuration possible
migration and long runs possible	(like host)

"A 3.10 Linux kernel is the minimum requirement for Docker"
- we have 2.6.32 :-(

→ Implementation of Docker containers is planned with RHEL 7.

Next steps

Installation issues...

- This week, phase 1 of our installation (130 TFlops) will be integrated into the newer system.
- New IP ranges are created with plenty of vacancies for virtual machines.

→ afterwards VM/gLite goes productive

Learn about the `State=CLOUD` stuff ... and possibly use it.

SchedMD and Slurm Community

Thank you for the great and fast support!