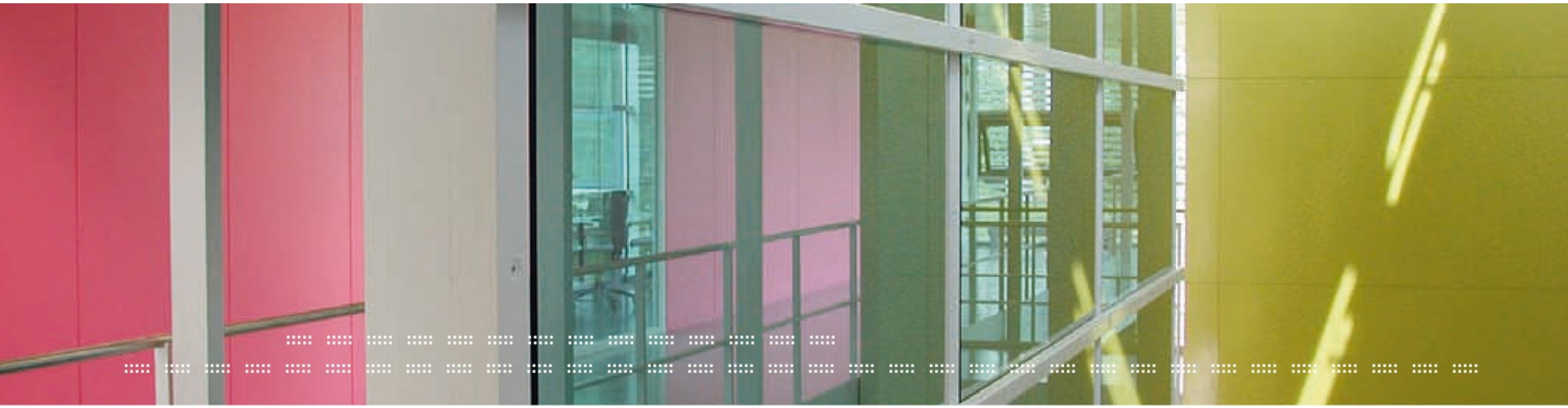
vTorque - Introducing virtualization capabilities to Torque

Nico Struckmann

Workbench on Sustained Simulation Performance (WSSP), Stuttgart, Oct 10th-11th 2017



Agenda

- Motivation
- Clouds
- Torque
- vTorque
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions
- Future Work
- Q&A

Motivation

- Today‘s HPC infrastructures are static
  - no choice to pick the most appropriate kernel
- High effort required to satisfy different user needs
  - i.e. modules system (different versions of a lib)
- Increasing complexity
  - maintaining many different combinations of modules (i.e. applying security patches)
- High effort for maintenance
  - to built a new image for compute nodes and to ensure it works as desired
- No fault tolerance
  - a single bad node can cause a (huge and long lasting) job to crash anytime

Motivation (2)

- Users/developers need to adapt their code to each HPC environment, i.e.
  - Mount paths
  - Workspace mechanisms
  - Batch system support
  - CPU and other hardware properties
- Developers cannot develop on commodity hardware and then run it without further adaptation in HPC environments

Agenda

- Motivation ✓
- Clouds
- Torque
- vTorque
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions
- Future Work
- Q&A

Clouds

- Flexibility
  - Guest is independent of host‘s kernel and libraries
  - Applications packaged as ready to run images
  - Contextualization for further customization
  - Built once, run everywhere (as long as the CPU arch matches)
- Fault Tolerance
  - Live migration
  - Suspend and resume
  - Checkpointing

Clouds (2)

- Less administration effort for maintenance of host systems required
  - No user related software installed on the host
- Performance
  - Computation is optimized (kvm kernel module)
  - Overhead is mostly due to I/O
  - No opensource virtual RDMA

Agenda

- Motivation ✓
- Clouds ✓
- Torque
- vTorque
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions
- Future Work
- Q&A

Torque

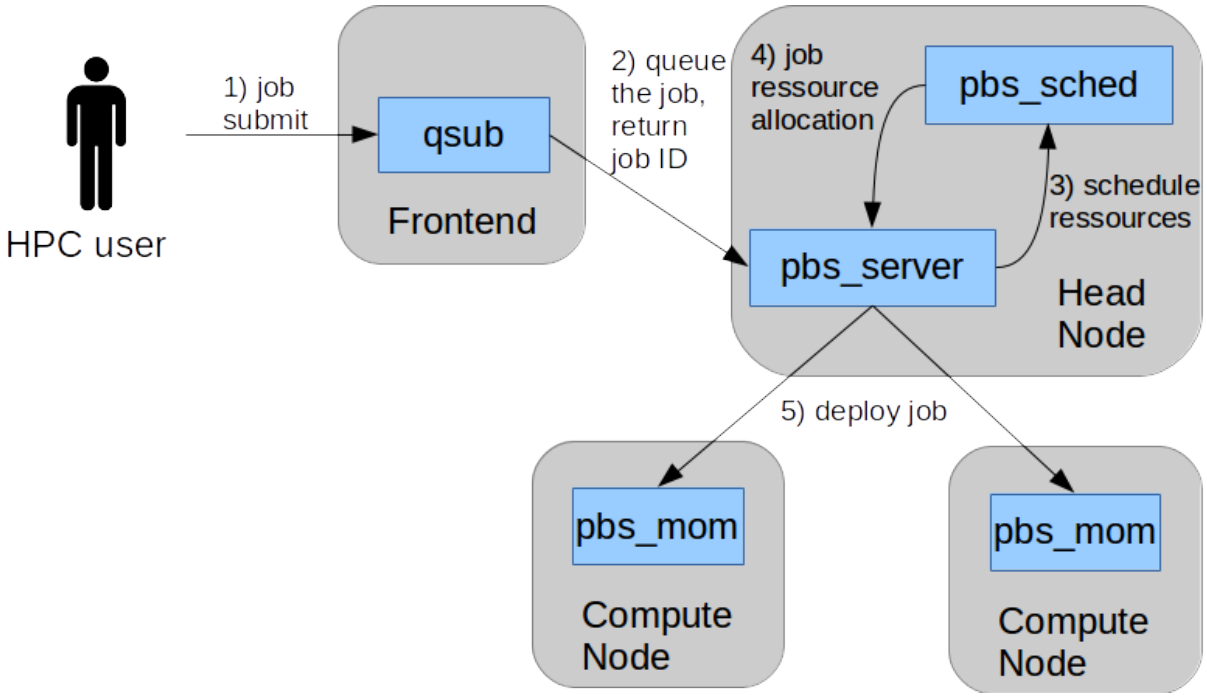
- Widely spread HPC batch system (resource manager and simple job scheduler)
- OpenSoure, derived from OpenPBS, developed by Adaptive Computing
- Provides control over batch jobs and distributed computing resources
- Supports sophisticated meta-scheduler
  - Moab/Maui
- No virtualization support, jobs are executed on bare metal nodes

Torque (2)

- Cli tools
  - qsub: command line job submission tool for users
  - qmgr: queue configuration for admins
- Main Components
  - pbs_server: central component
  - pbs_sched: schedules jobs on resources
  - pbs_mom: compute node daemon

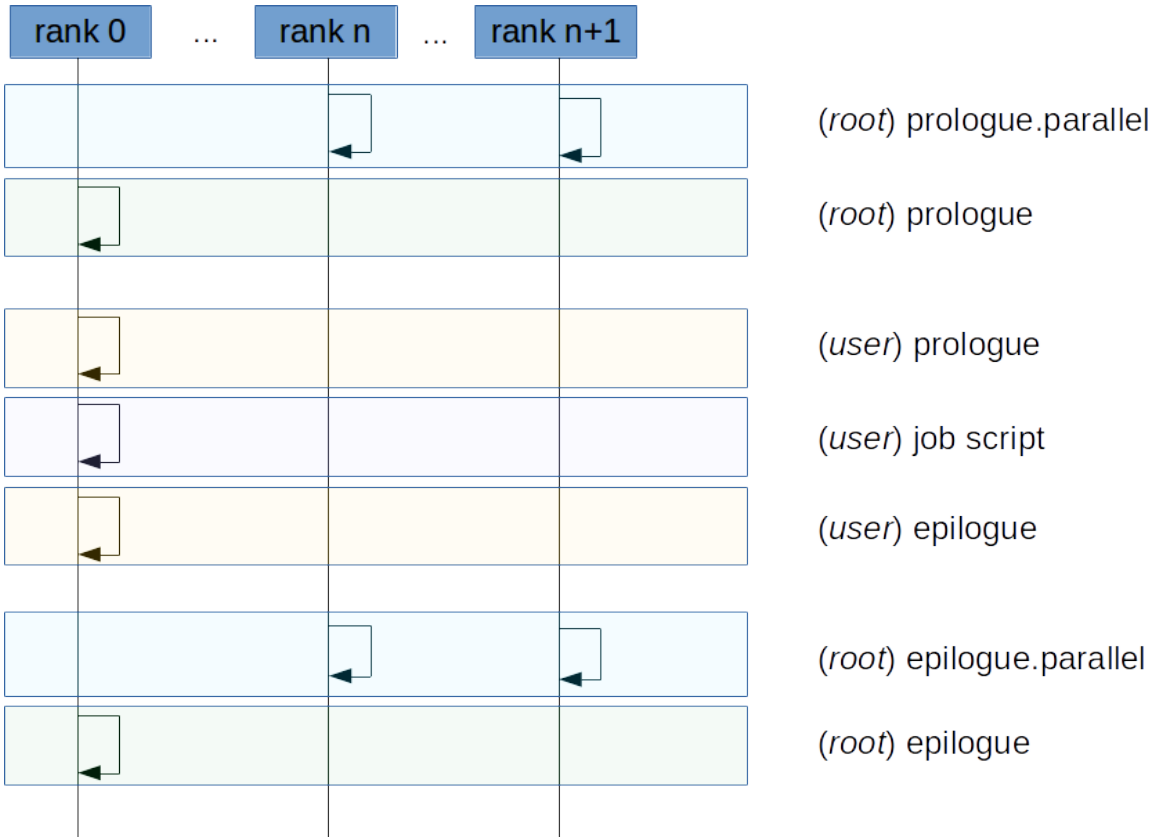
Torque (3)

- Workflow: Users submit their batch jobs, Torque queues, schedules and deploys them on allocated compute nodes



Torque (4)

- Provides capabilities to run preparation and clean up tasks (prologue, epilogue)



Agenda

- Motivation ✓
- Clouds ✓
- Torque ✓
- vTorque
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions
- Future Work
- Q&A

vTorque :: Overview

- Combines the flexibility of virtualization with Torque
- Set of **non-intrusive** wrapper scripts and templates, written in bash
- Comes with two new CLIs:
  - vsub: counterpart of qsub, for virtualized jobs
  - vmgr: manage and list available VM images
- Admins can define default values (`vsub <jobscript>`)
- Admins can en/disable features and functionality
- User can override defaults at job submission time
- Enables dual use of computing resources

vTorque :: Features

- Deploys job script transparently in virtual environment
- Abstraction layer for environment specific properties
  - i.e. mount points (`/home`, `/workspace`)
- Administrators define min/max and default values
  - i.e. for IOCM, vCPUs
- vCPU pinning supported
  - Custom mapping file or automatic pinning by numad
- Bare metal nodes and virtual guests share (network) filesystems
- Cluster‘s `/opt` is available as `/opt-hpc` in guests to allow applictions to make use of proprietary libs (if binary compatible)

vTorque :: Features (2)

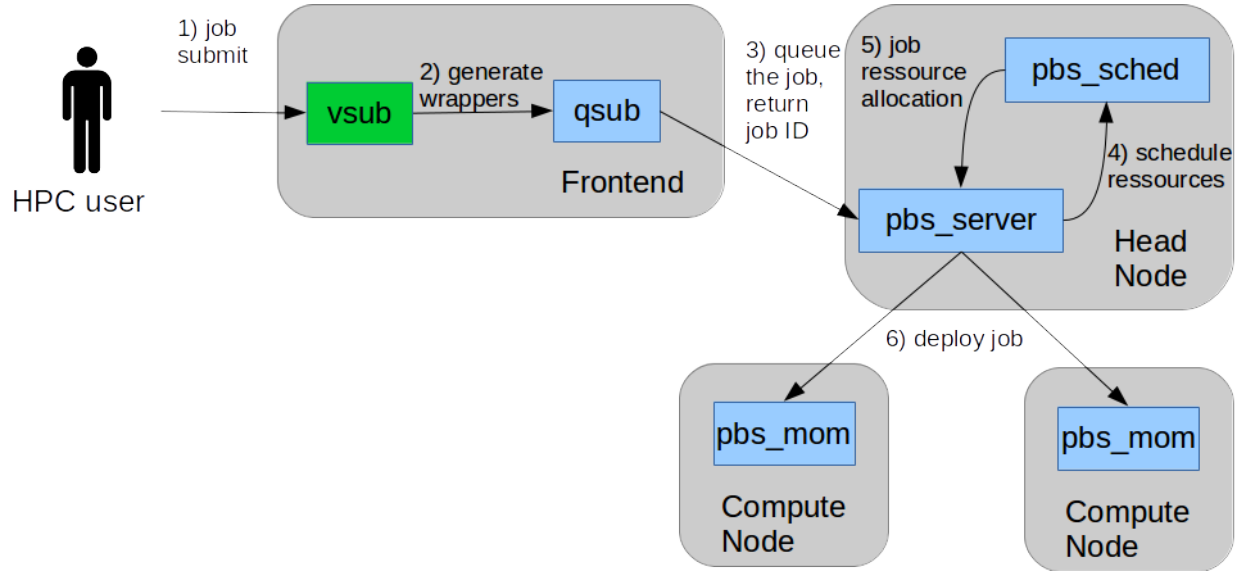
- Multiple VMs per node supported
- Supports guest OS
  - standard Linux guests
  - and Unikernel OSv
- Contextualization and costumization via NoCloud metadata
- Root and user level VM pro/epilogue script support (for standard Linux guests)
- Mixed resource requests (coming soon)
- HPC Network abstraction
- Several optional components available
  - i.e. for resource usage monitoring (host to app level) or logging

vTorque :: Deployment

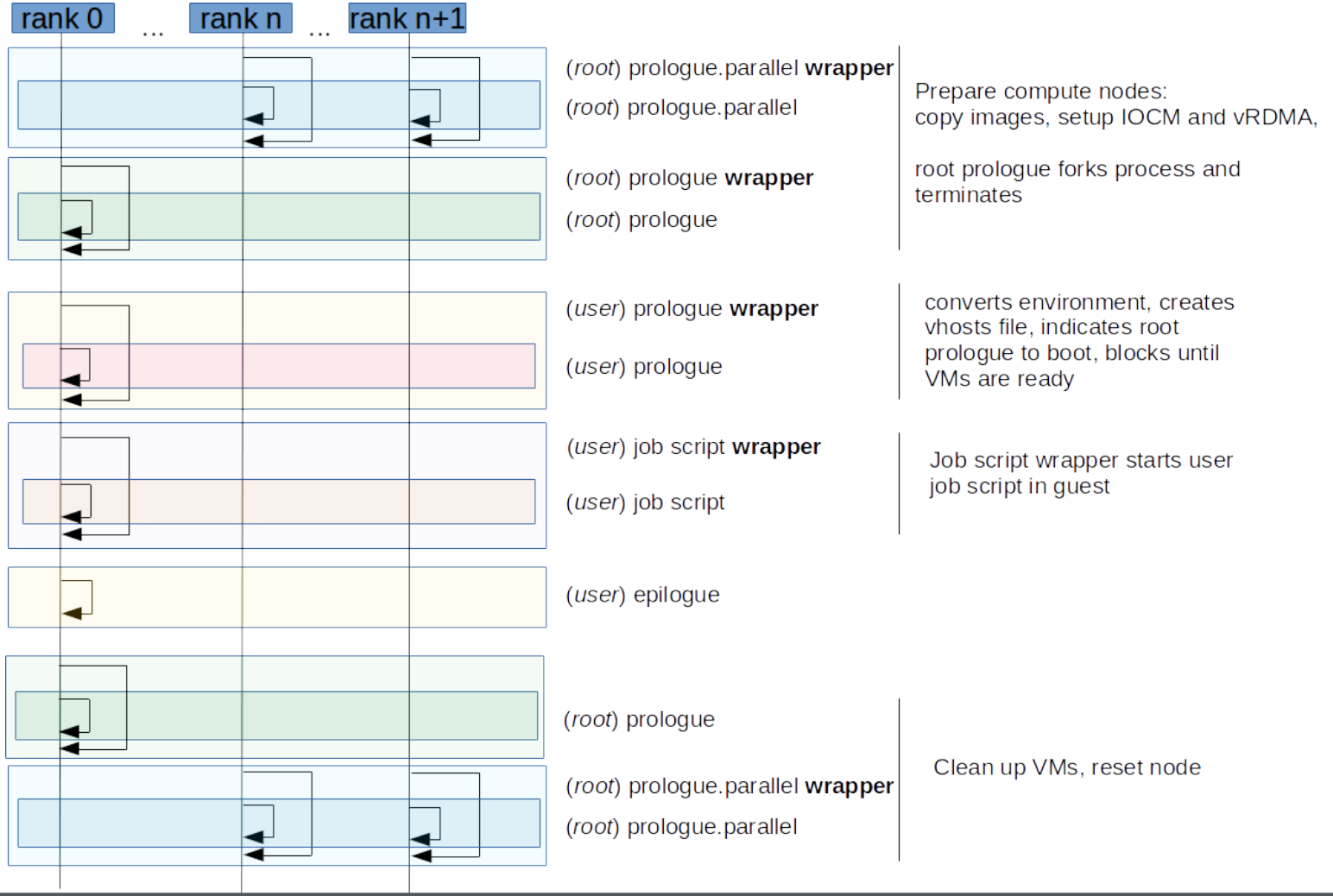
- Requires working Torque installation
- Requires working (s)KVM installation on compute nodes
- Setup script using pdsh available
  - Increase pro/epilogue timeout (default 30 sec)
  - adds vTorque CLIs (vsub,vmgr) to `$PATH`
  - Renames root level scripts in `/var/spool/torque/mom_priv/{pro,epi}logue{.parallel}` to `*.orig`
  - Root level wrapper scripts are put into place as `/var/spool/torque/mom_priv/{pro,epi}logue{.parallel}`

vTorque :: Workflow

- **`vsub`** generates wrappers for user prologue
  - `qsub -l prologue=<user prologue wrapper>`



vTorque :: Workflow



vTorque :: Security and Data Privacy

- Users cannot be granted to boot VM images directly
  - They may upload an image, boot it, gain root, change uid and access other users private data on NFS
- Users cannot be granted to use own images
  - May provide them root access
- Root prologue needs to ensure generated files cannot be modified by the user
  - Users may exploit them, i.e. add root user to metadata file, mount another users home/workspace, install vulnerable packages
  - Achieved by passing on parameters to root prologue to generates the metadata file and keep it readonly
- Virtual Nodes
  - Exclusively allocated for one user per default
  - user‘s home is mounted (e.g. ‘`/home/user`‘ instead of ‘`/home`‘)

vTorque :: Optional Components

- Snap-Telemetry (INTEL)
  - Open telemetry framework designed to simplify the collection, processing and publishing of system data through a single API
  - Offers plug-ins for diverse metrics
  - Graphana backend for visualization of collected data
- vRDMA (HUAWEI)
  - Open source virtual RDMA functionality
  - Allows to provide one physical IB card to multiple guests
  - Close to bare metal performance
- IOCM (IBM)
  - IO Core Manager
  - virtual I/O performance superior to the vanilla KVM hypervisor
  - Hypervisor schedules I/O processing, instead of Linux thread scheduler, reduces overhead of thread context switching

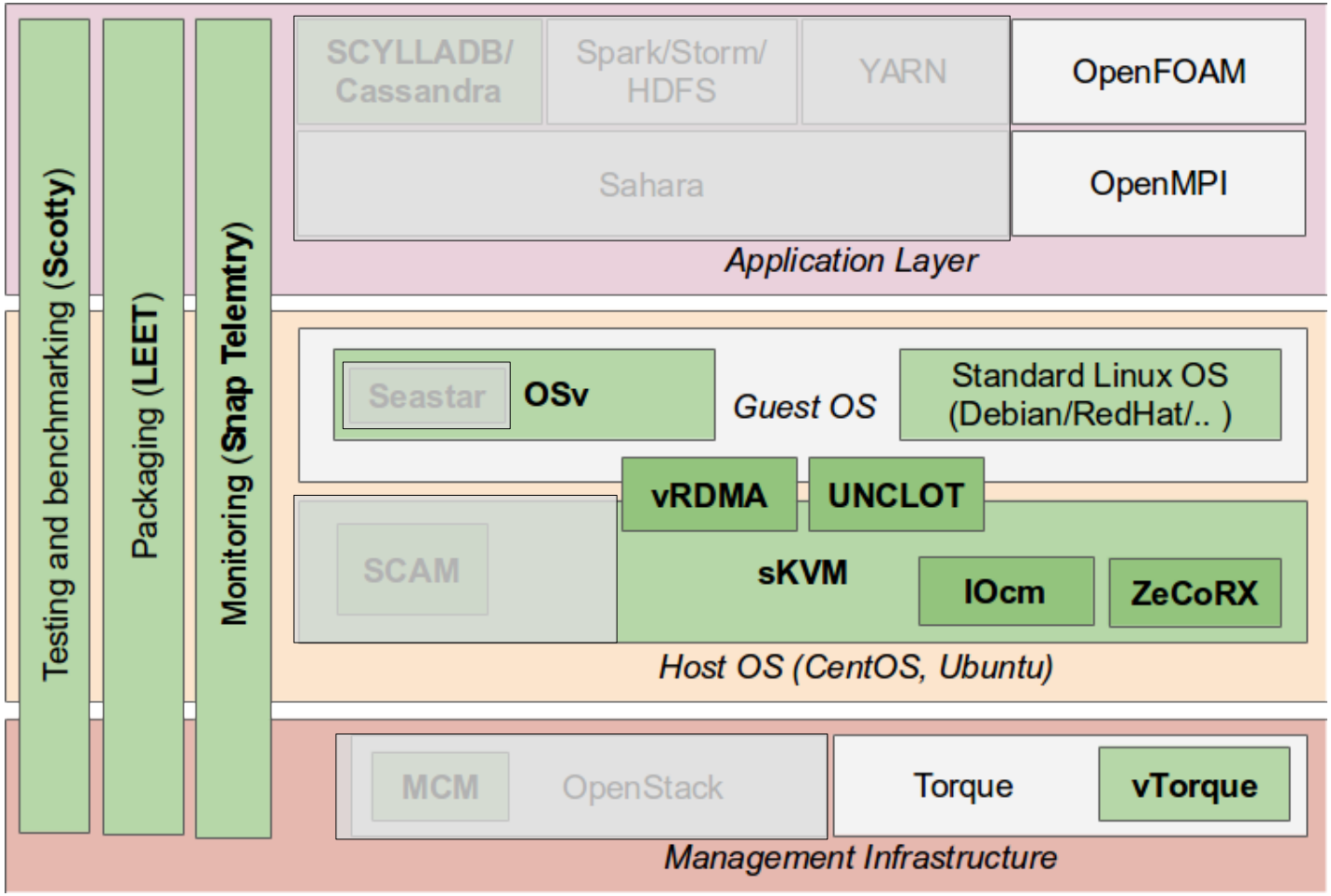
vTorque :: Optional Components (2)

- UNCLOT (XLAB)
  - Shared memory between VMs on same host
- OSv (SCYLLA)
  - Small images
  - Fast boot times
  - Single user
  - Single process
  - No context switches needed
- LEET (XLAB)
  - OSv Application packaging (VM image building)

vTorque :: Optional Components (3)

- Scotty (GWDG)
  - Automated experiment execution and CI
- Log4bsh (HLRS)
  - Logging facility for vTorque
  - Formerly part of vTorque, meanwhile separate project
  - Highly configurable (log level, log file, logging behaviour, hooks, ..)
  - Logs beside the msg, log level and timestamp also executing script‘s name and host

Overall HPC Software Stack



Agenda

- Motivation ✓
- Clouds ✓
- Torque ✓
- vTorque ✓
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions
- Future Work
- Q&A

Conclusions

- Reducing I/O virtualization overhead is crucial for HPC adpotion
  - with sKVM were on a promising way as it seems
- Applications with different workload characteristics may be deployed on the same node (on different Numa domains) to increase overall resource utilization
  - i.e. I/O-bound and CPU-bound may match
- HPC environments can be improved for all stakeholders by introducing flexibility through virtualization and packaged applications
  - Owner: new user groups with conflicting requirements can be served
  - Admin: reduced maintenance effort and increased security (lean host and guest OS), Linux guests can apply automatic security updates at boot time, convenient application packaging
  - Developer: can develop on commodity hardware and built HPC ready applications, ship it with an optimized predefined environment
  - User: more flexibility, no longer restricted by the compute node‘s kernel and software

Conclusions (2)

- Copying VM images during prologue sequence
  - takes additional time, bandwidth and space
- Waiting for guests to become ready
  - may take some time if the image is outdated and unattended updates are applied during instantiation
- Issues in prologue can render all nodes unavailable
- Additional IP addresses needed for guests
- Suspend and Resume must be supported on higher levels i.e. MPI layer

Agenda

- Motivation ✓
- Clouds ✓
- Torque ✓
- vTorque ✓
  - Overview
  - Deployment
  - Features
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions ✓
- Future Work
- Q&A

Future Work

- Global spare node mgmt for Torque
- Automatic live migration, if nodehealth check indicates degrading state
- Generic PCIe passthrough to support i.e. accelerators
- Stripped down host OS
  - i.e. Tiny Core OS
- Implement PoC wrapper solution as C-patch for Torque
  - Better performance for parsing files, arguments, etc
  - No temp files in user space required
- Suspend and Resume capabilities for MPI
- Interactive VM jobs

Agenda

- Motivation ✓
- Clouds ✓
- Torque ✓
- vTorque ✓
  - Overview
  - Features
  - Deployment
  - Workflow
  - Security and Data Privacy
  - Optional Components
  - Overall HPC Software Stack
- Conclusions ✓
- Future Work ✓
- Q&A

Q&A

src: https://pixabay.com/p-1015308
license: CC0 Creative Commons

Acknoledgements

- vTorque has been developed within project **MIKELANGELO**

MIcro **KE**rne**L** virtualiz**A**tio**N** for hi**G**h p**E**rformance c**LO**ud and hpc systems

Co-funded by the Horizon 2020 Framework Programme of the European Union

- RIA project no. 645402
  - Started in January 2015
  - Ends in December 2017
  - Co-funded by the European Commission under H2020-ICT- 07-2014: Advanced Cloud Infrastructures and Services program

src: https://www.flickr.com/photos/wwworks/4759535950
license: Attribution 2.0 Generic (CC BY 2.0)