University of Glasgow

THE UNIVERSITY of EDINBURGH

Evaluation of containers as a virtualisation alternative for HEP workloads

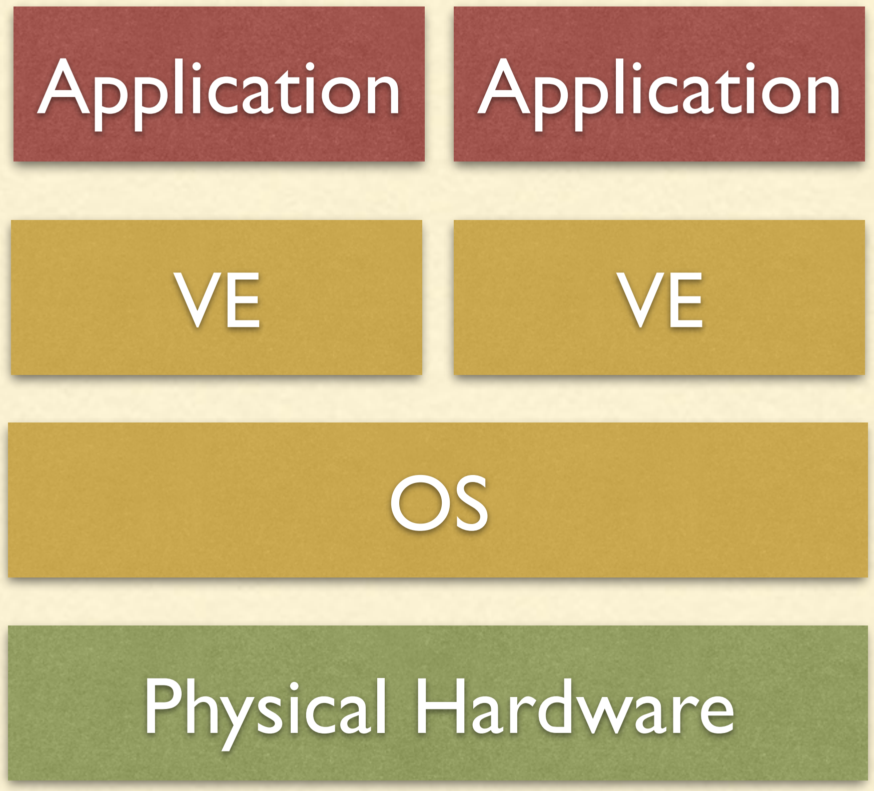
Gareth Roy[1], **Andrew Washbrook**[2],
David Crooks[1], Gordon Stewart[1], Gang Qin[1], Samuel Skipsey[1], Dave Britton[1]
[1]University of Glasgow [2]University of Edinburgh

21st International Conference on Computing in High Energy Physics
13th April 2015

Outline

- What are Containers?
- The Container ecosystem
- Container deployment and management
- HEP benchmarking
- Conclusions

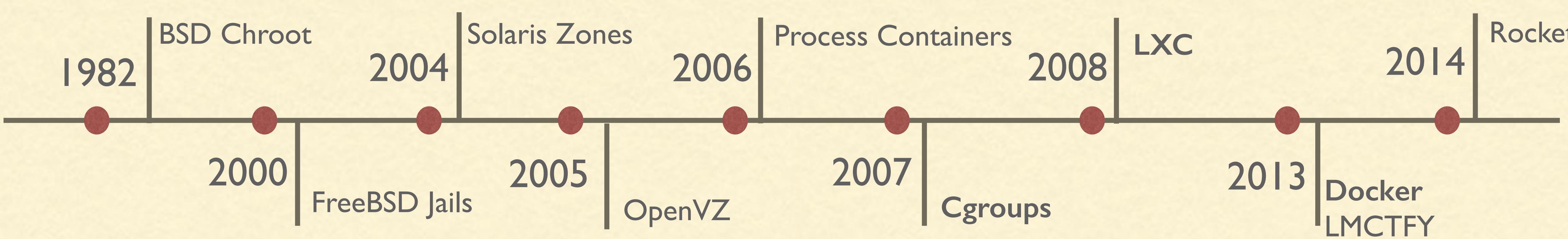
Containerisation



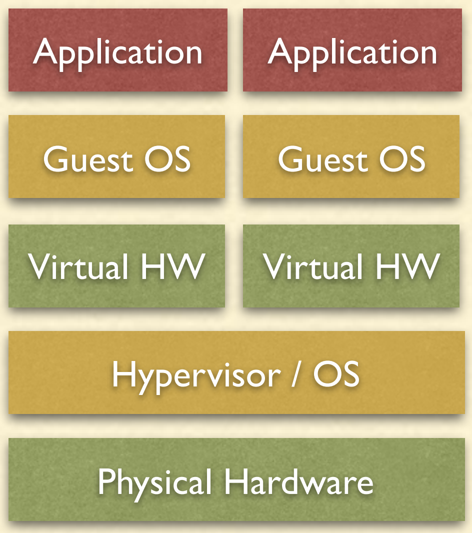
Linux Container

- Containerisation is a form of **OS level virtualisation**
- The Linux kernel hosts multiple partitioned user-land instances (Virtual Environments)
- Accomplished through separate *namespaces* for filesystem mounts, network, processes and users
- Backing storage can be Copy-on-Write or a union filesystem (UnionFS/AUFS)

Containers Timeline

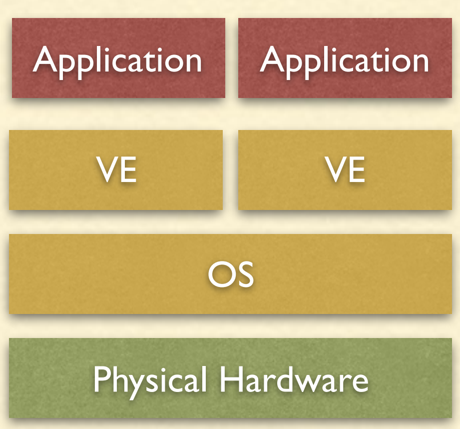


Comparison with Virtual Machines



Virtual Machine

- **Pros:**
  - OS independent
  - Security Model
  - Live migration
  - Mature ecosystem
- **Cons:**
  - Full system image
  - Slow startup and build
  - Memory consumption
  - Opaque to host



Linux Container

- **Pros:**
  - Low barrier of entry
  - Fast Instantiation
  - Native Performance
  - Deployment Flexibility
- **Cons:**
  - Restricted to Linux
  - Shared Kernel
  - Security Model
  - Young Ecosystem

The Container Ecosystem

- There is a growing ecosystem of tools and services to deploy, manage and orchestrate containers
- The most popular container application platform is **Docker**

LXC - tools for container lifecycle management

Open source minimal OS specifically designed to host and cluster application containers

Core OS

Application virtualisation engine based on containers (*LXC, libcontainer*)

docker

Kubernetes - Docker container orchestration system for large scale application deployment

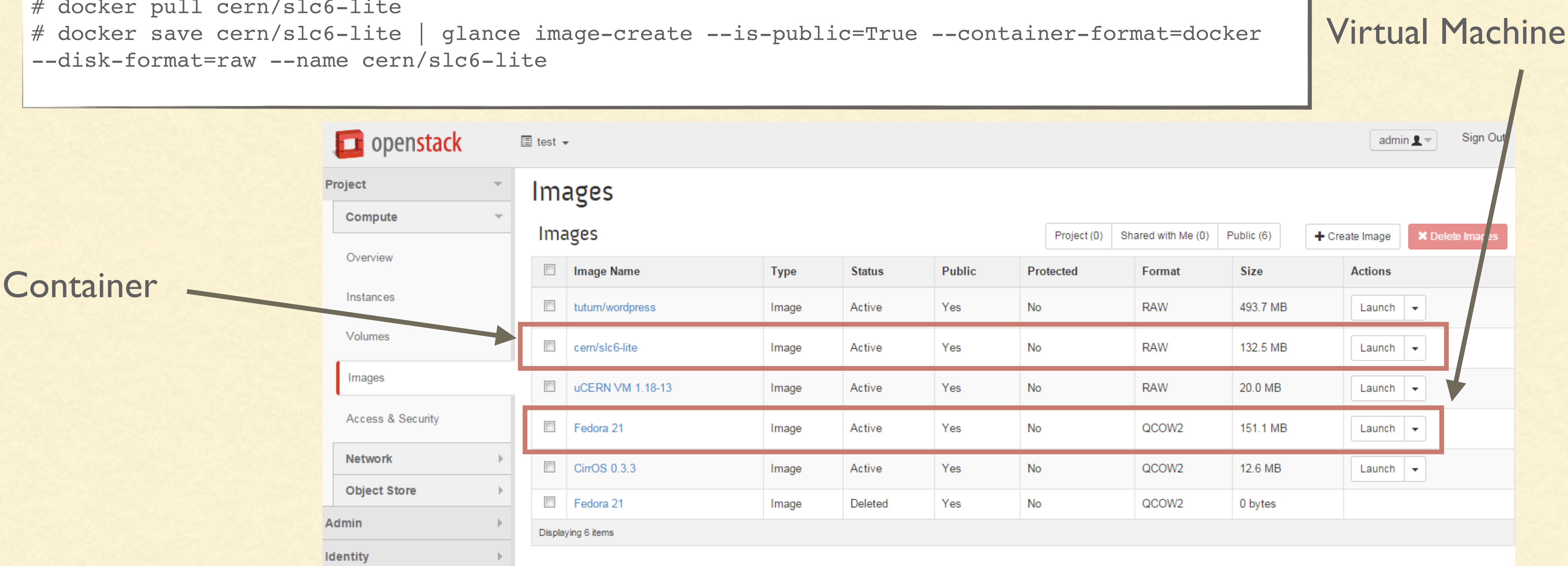
Tools for working with containers focusing on the *Application Container Image* (an open standard for container formats)

Rocket

Openstack Container Management

- Explored the readiness of OpenStack to natively support the management of containers
- Enables easy integration with cloud infrastructure available in WLCG
- A Docker driver is not in the current Openstack release (Juno)
  - Manually install driver: https://wiki.openstack.org/wiki/Docker

```
# docker pull cern/slc6-lite
# docker save cern/slc6-lite | glance image-create --is-public=True --container-format=docker
--disk-format=raw --name cern/slc6-lite
```



HEP Containers

Container comparison with CERNVM image

- Pull base CentOS 6 image from Docker registry
- CVMFS mounted as an external volume
- Increase storage in the container for datasets and job output

CVMFS integration

- CVMFS requires root-privileges for FUSE interaction.
- *Either* run a **privileged** container and export the CVMFS volume to other containers
  - Security implications
- *Or* export the CVMFS volume from the host
  - Not a flexible hypervisor solution
  - Can lead to issues with other container management tools (CoreOS, Project Atomic)

HEP Workload Performance Testing

Motivation

- Do containers offer native performance for realistic HEP-based workload?
- What is the performance penalty for using Virtual Machines over bare metal (and containers)?

Workload types

- HEPSPEC benchmark
- Geant4 Monte Carlo Simulation
- Event Reconstruction
- Monte Carlo event generation

HEP Workload Performance Testing (2)

Test Platform

- Run each HEP workload type on two testbed servers; one containing an Intel Xeon processor and the other an Avoton processor
  - **Avoton**: Low power Atom-based 22nm SoC device
- Run each workload type on bare metal, in a virtual machine (KVM) and in a container (Docker)
  - Adapted a µCERNVM image for testing purposes
  - Tested both a RAW image file and a LVM partition as VM backing storage

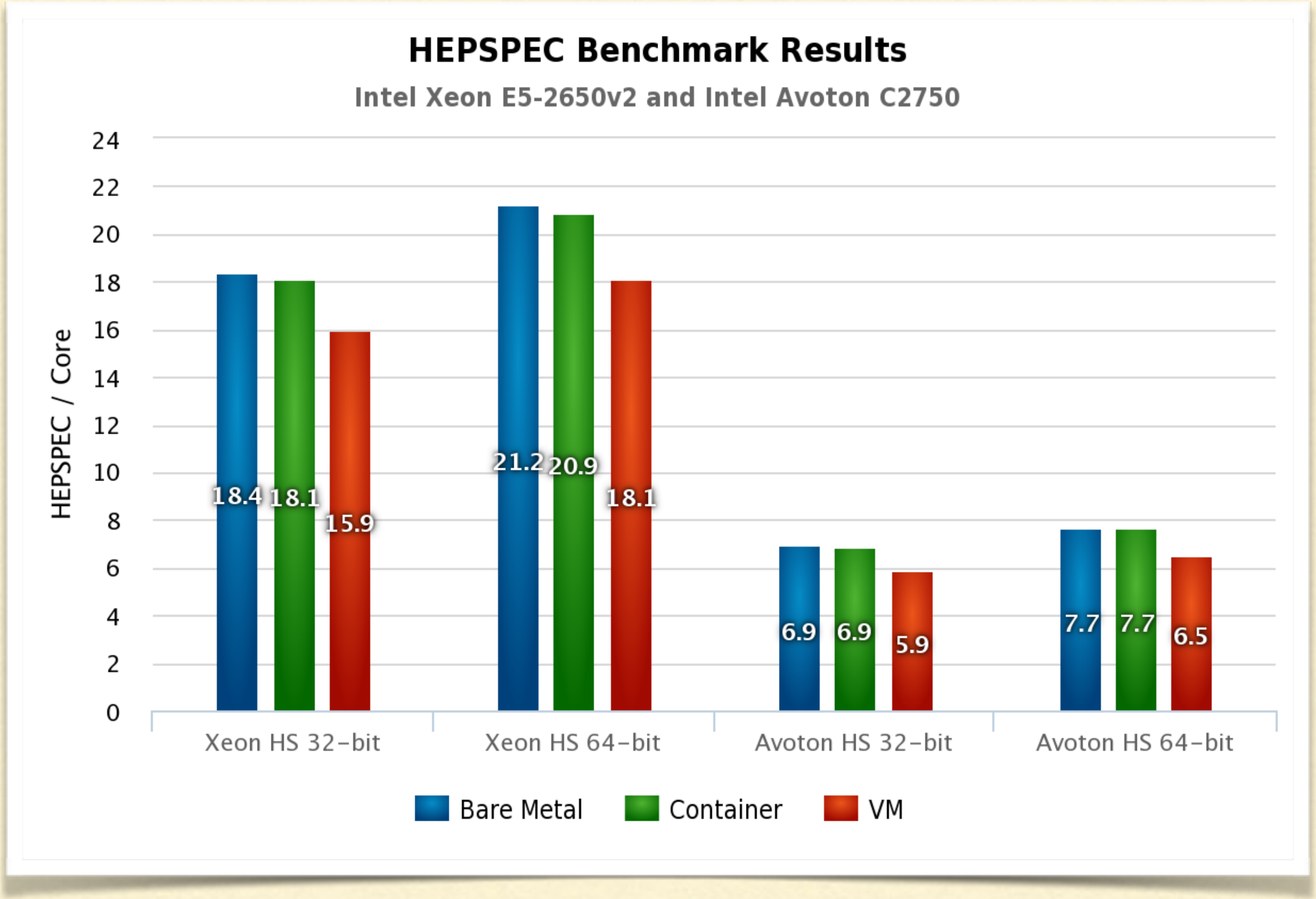
- SuperMicro Twin Squared
  - 2 x E5-2650 v2 2.6GHz
    - 16 cores (32 Hyper threaded)
  - 64 GB RAM
  - 2 x 500GB 7200rpm SATA 6.0

- DELL FX2 FM120 Avoton
  - Intel C2750 2.4Ghz
    - 8 cores (Avoton Atom CPU)
  - 16GB RAM
  - 80GB SSD

Test Patterns

- Run each test multiple times to validate performance and timing consistency
- Run single core and *N* (= number of cores) simultaneous workload instances

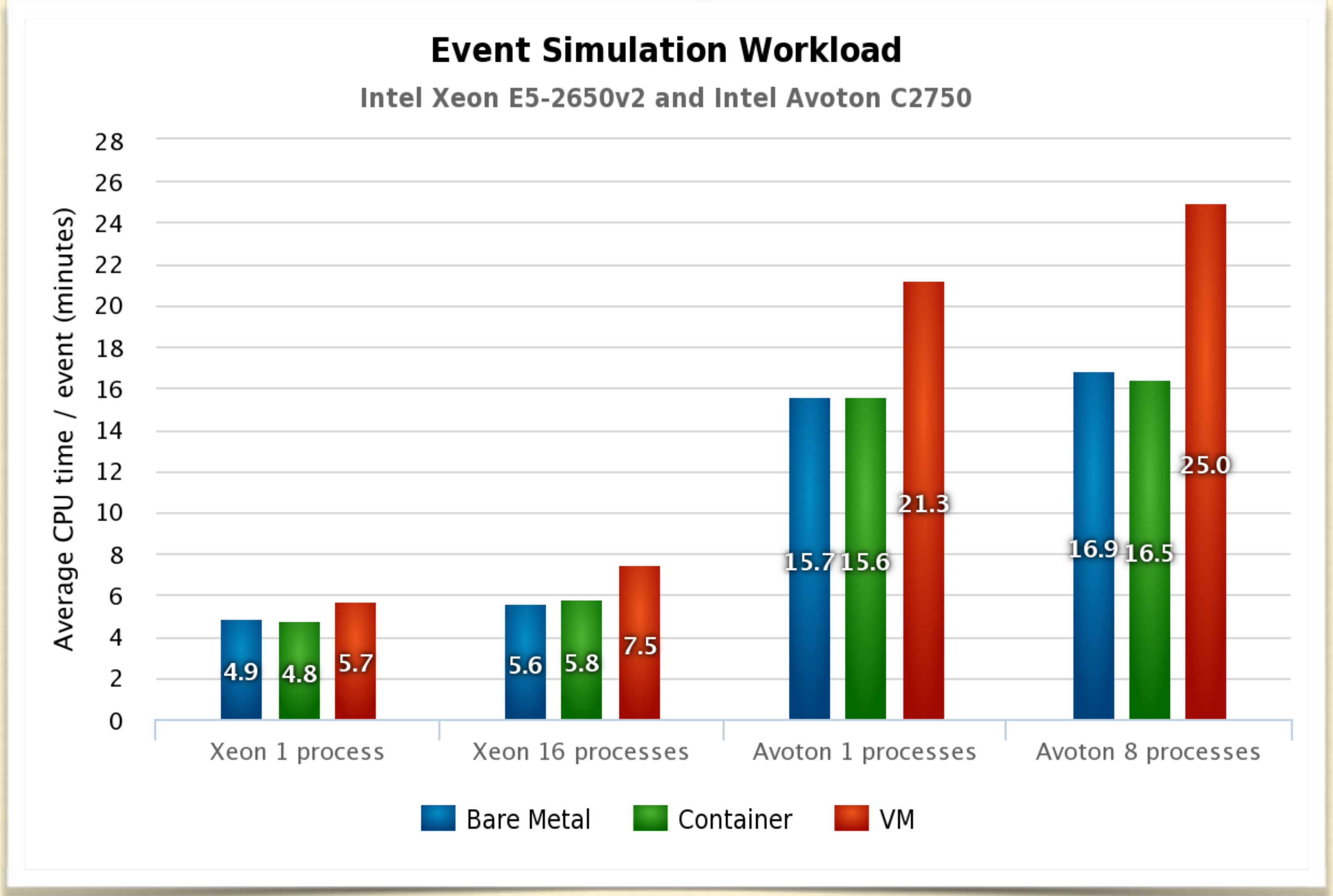
HEPSPEC Benchmark Results



- Containers are within 1% of "native" HEPSPEC performance
- Benchmark score for VMs are **14.7%** less for the Xeon and **15.3%** less for the Avoton compared to the native HEPSPEC (64-bit) score

MC Simulation Results

- Focus on relative performance in event processing loop
- CPU time/event (as reported by application) is averaged over 40 events



- Containers demonstrate near-native performance
- Single process MC simulation timing performance is **13.4%** lower for the Xeon and **26.4%** lower for the Avoton
- For 16 (8) simultaneous processes: VM is **24.7%** lower for Xeon (**32.5%** lower for Avoton)

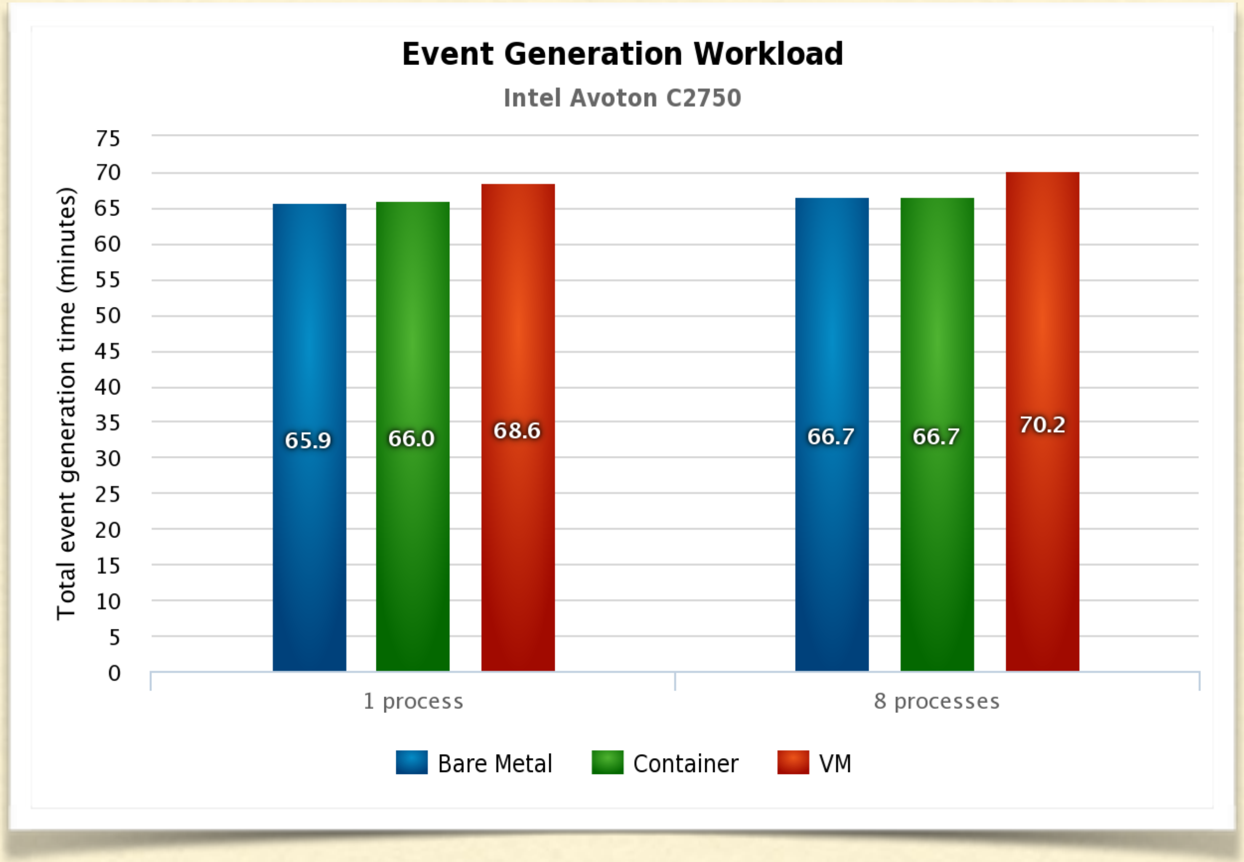
Event Generation and Reconstruction Results

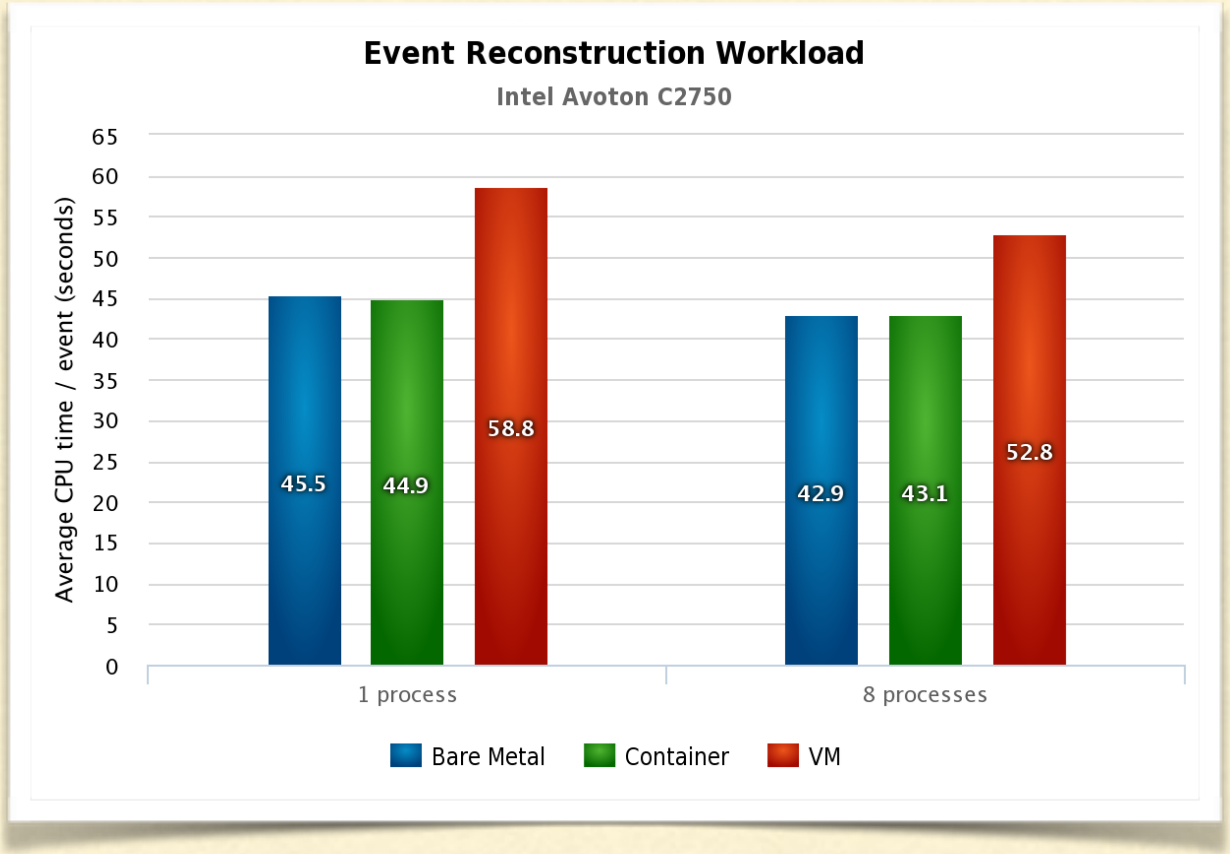
Event Generation

- Measured total time taken to generate large sample of $Z\mu\mu$ events
- VMS lose only **5%** performance compared to bare metal and containers

Event Reconstruction

- CPU time/event averaged over 50 events
- VMs show **22.6%** performance drop for a single process, **18.8%** for 8 simultaneous processes





Conclusions

- Containers are a compelling alternative to whole system virtualisation
- The performance of containers for HEP workloads are similar (if not the same) as native execution
- Virtual Machines observed to give a reduction in performance of 15-20% on HEP workloads
  - Caveat: VMs could be tuned to give better performance

Future Work

- Potential deployment of containers on existing HEP cloud resources
- Does HEP lend itself to the single application model preferred by Docker?
  - Consider: middleware deployment, distributed computing components (e.g pilots)
- Effort has already started (see next presentation!)

Any Questions?