
CERN

HEP-Benchmarks & CERN Batch profiling

Luis Fernandez Alvarez

HEPiX Benchmarking Working Group (17/01/2020)

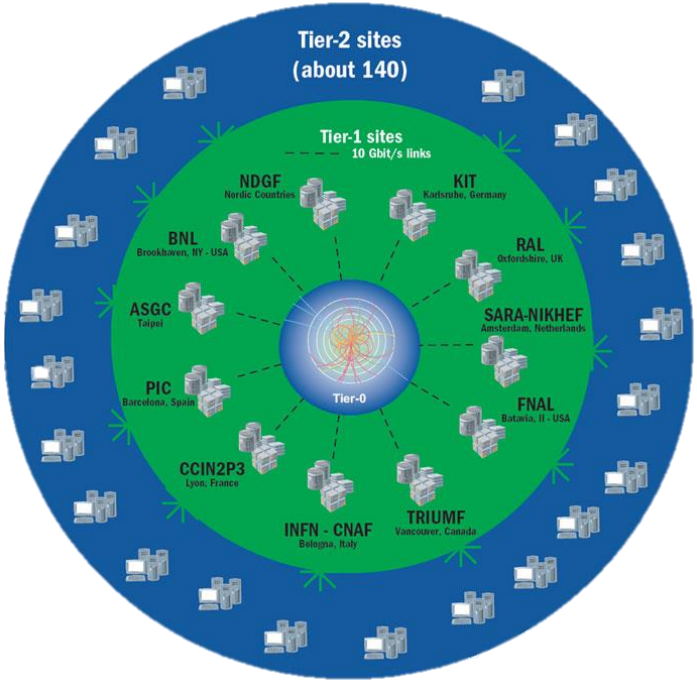
Why HEP-Benchmarks in Batch?

The Batch Service @ CERN IT

Provides Tier-0 compute power via HTCondor to WLCG.

- Process CPU intensive workload ensuring fairshare among various user groups
- Maximize utilization, throughput, efficiency
- It runs jobs from the Grid and from local CERN departments

235K	20K	1M
Total Cores	Virtual Machines	Completed / Day



Heterogeneous hardware

- Our HTCondor cluster is a heterogenous pool of resources.
  - Different hardware with different configuration
    - VMs/Baremetal, SLC6/CentOS7, SSD/HDD, Kernels, storage…
  - Hosted in different datacenters: Meyrin, Wigner, LHCb containers, external clouds,…

HEP-Benchmarks scenarios

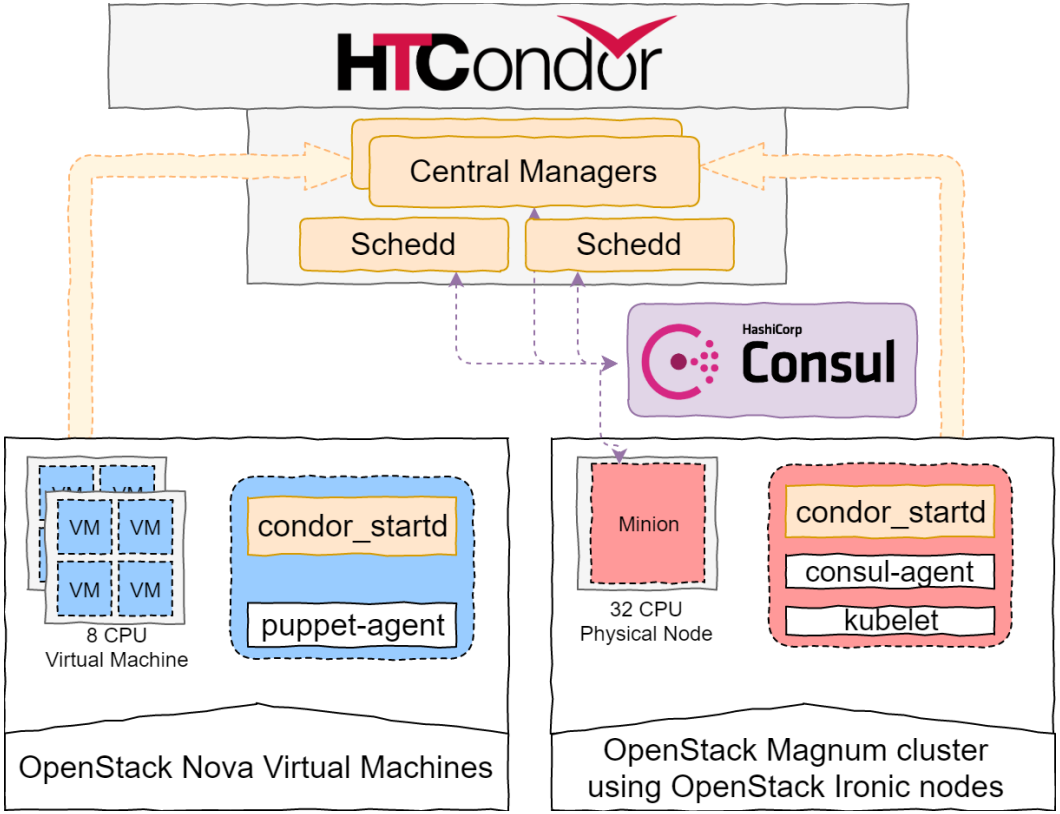
- We are **not** focused on benchmarking the HW under ideal circumstances.
  - Procurement / Cloud team dealing with that.
- Benchmarking in a setup similar to production, or in production during the life-cycle of the hardware.
- Adopt benchmarks to measure the impact of our heterogeneous configuration:
  - *Procurement says our HW performs X, but we get Y in datacenter Z. Can we do better?*
  - *VM scored X for the last months. It know scores Y. What changed?*
  - *How much could we gain running using XYZ?*
  - *How expensive is to run benchmark X in cloud Y?*

Use case: from Puppet VMs to k8s baremetal

CHEP 2019: Managing the CERN Batch System with Kubernetes

Prototype

- Hybrid HTCondor cluster with worker nodes in two forms:
  - Traditional Puppet manged VMs
  - Kubernetes based minions
- New element: Consul.
- Total capacity of 100 x 32CPU (SMT-ON) machines:
  - 50%: Puppet 8 CPU virtual machines
  - 50%: Kubernetes baremetal nodes
- Many areas of interest to evaluate:
  - Operations impact
  - Resource usage efficiency

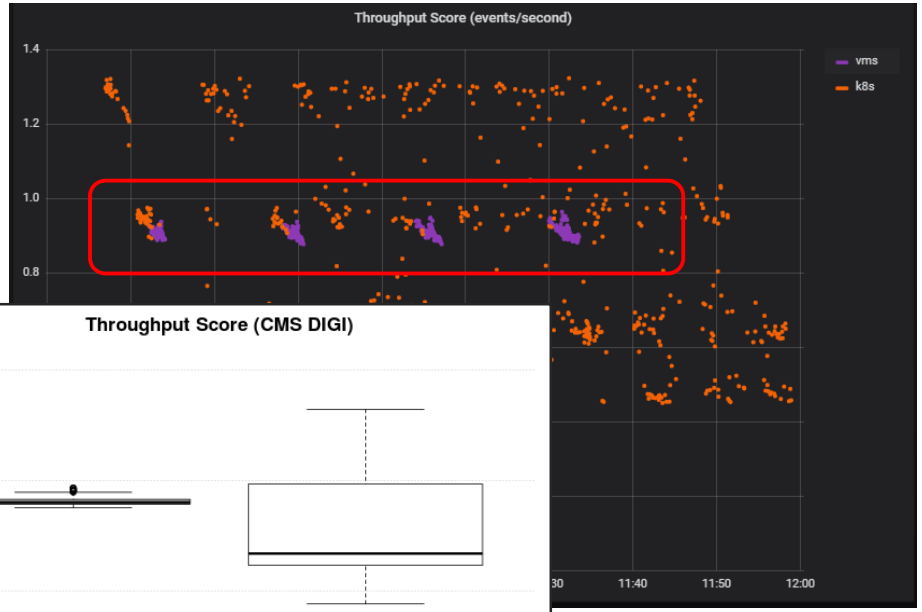


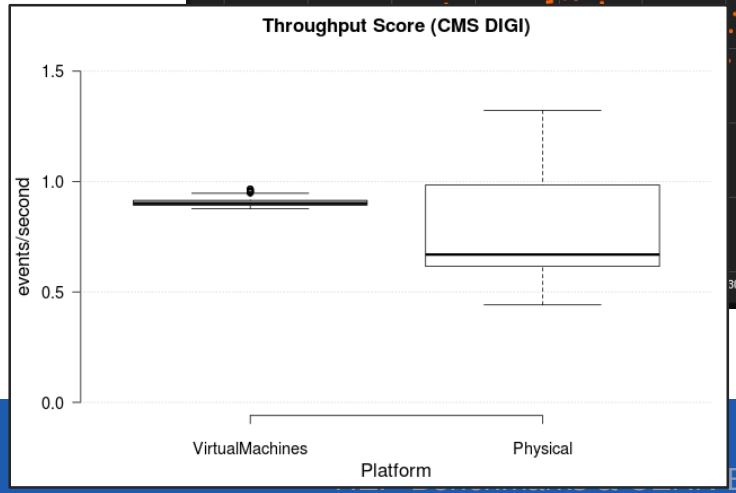
Benchmarking

- Evaluate performance of both models
- Benefit from the current effort of the Benchmark WG: hep-workloads.
- Benchmarks submitted as HTCondor jobs:
  - 1600 cores per platform, CentOS7 workers.
  - 8 core jobs. Benchmark payload depending on the benchmark:
    - Single-threaded: 1 thread x 8 copies
    - Multi-threaded: 8 threads x 1 copy
  - 800 jobs per platform (VMs vs Kubernetes): resources filled 4 consecutive times
  - Mainly executed as Singularity jobs (SLC6 based benchmarks)
- Results sent to the CERN IT monitoring infrastructure to be indexed in ElasticSearch and visible via Grafana

Unexpected results

- Very spread results on baremetal.
- Total throughput lower in some benchmarks:
  - 800 Jobs.
  - VMs 1.5 times faster.
- Configuration review…



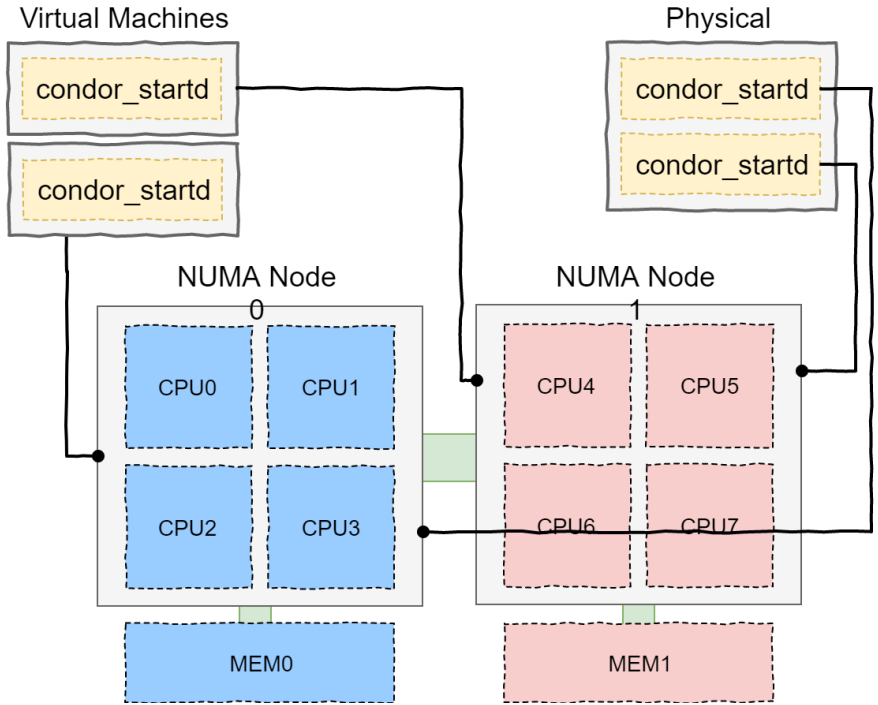


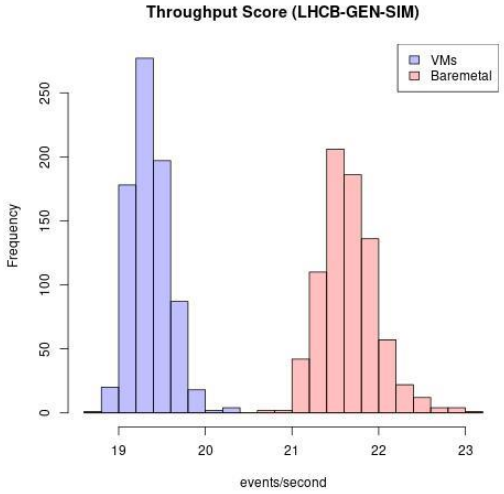
NUMA topology & scheduling

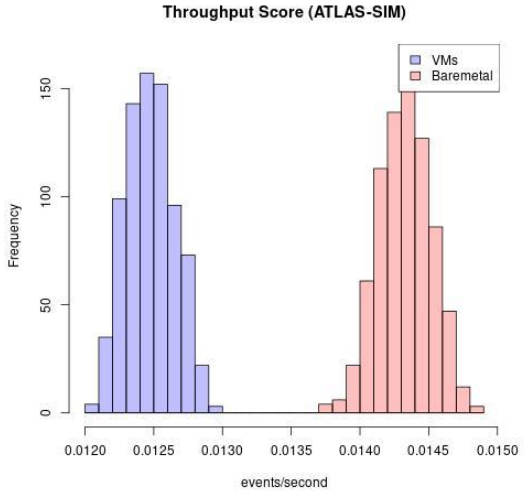
- Running a full-node benchmark (as defined in previous slide) in baremetal lower than equivalent VM setup.
- VMs are pinned to NUMA nodes.
- Kernel is not perfect scheduling, it can benefit from some hints.
  - Running two half-node instances pinned to NUMA nodes showed expected results.
- (How could/Should) these hints be expressed in the benchmarks?
  - Discussion opened in: https://its.cern.ch/jira/browse/BMK-261.

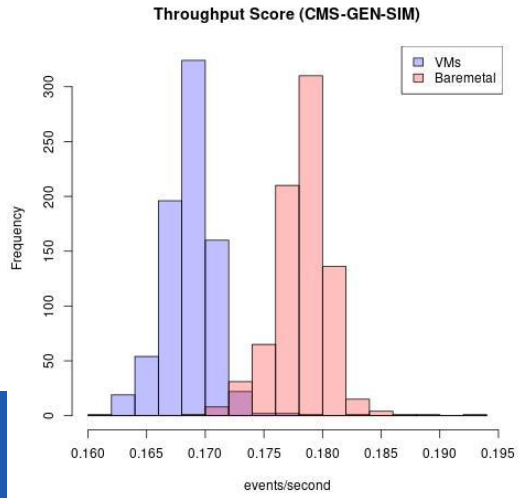
NUMA aware setup

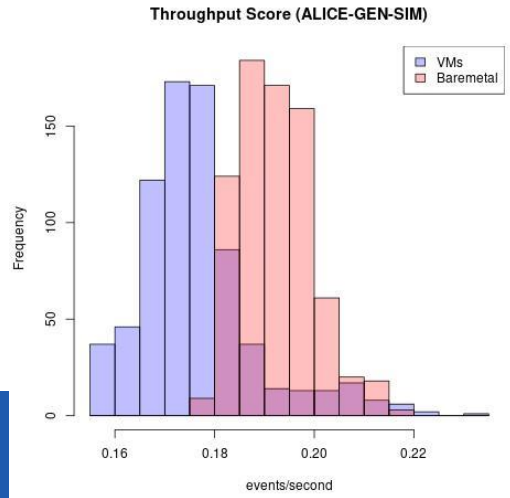
- We have to help the kernel scheduling by pinning processes to NUMA nodes.
  - Already solved in the CERN cloud by scheduling VMs to NUMA nodes: Optimisations of the Compute Resources in the CERN Cloud Service.
- `condor_startd` on VMs: automatically tied to NUMA node as VM already is
- Apply same principle to `condor_startd` on physical nodes: instantiate one daemon per NUMA node
- Use cpusets to confine each daemon
- Exposed via HTCondor as multiple slots:
  `slot1@numa0@<hostname>`
  `slot2@numa1@<hostname>`



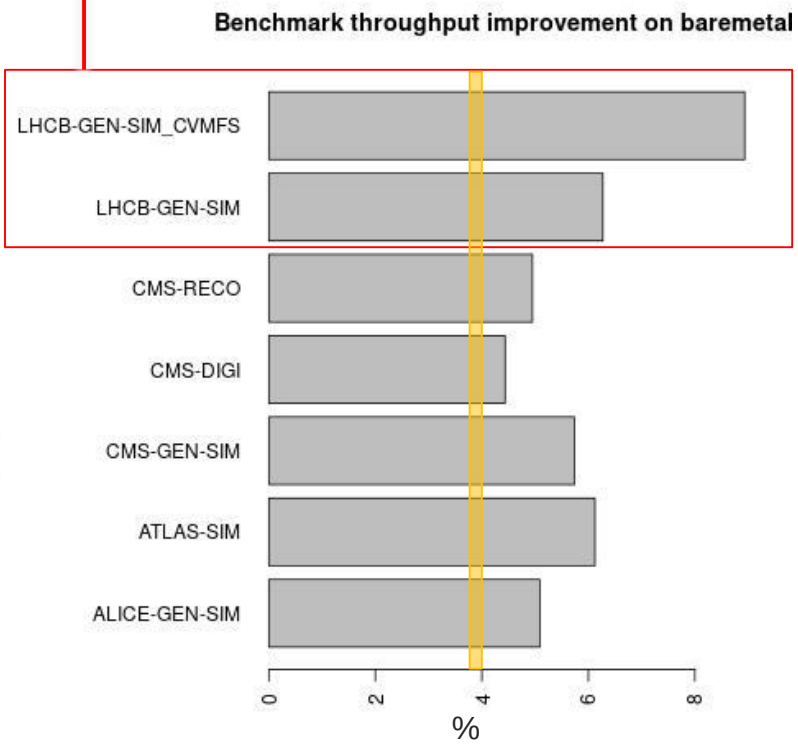








Difference between benchmarking in Singularity image vs accessing CVMFS? to be explored…



Conclusions

- HEP-Benchmarks proved to be useful.
  - It helped spotting potential inefficiencies and measuring performance improvements.
  - Little effort required from our side to start using them.
- Things nice to have:
  - Containerized benchmarks without cached data.
    - See impact of networking fetching data from CVMFS.
  - Ability to run on CentOS7 without depending on containers.
  - Publish them as CVMFS images in /cvmfs/unpacked.cern.ch?

Next steps

Next steps

- Automated Batch benchmarking:
  - Run benchmark continuously in all the Batch compute flavours:
    {vm|bare}-{scl6|cc7}-{qa|prod}
  - Store data and monitor.
  - See evolution over time for same machine.
  - Measure the impact of new OS, configuration, location,…
- Use benchmarks to measure job metrics in external clouds.

Thank you