CERN Report:
Batch farm worker nodes

Luis Fernández Álvarez <luis.fernandez.alvarez@cern.ch> - IT-CM-IS

13/07/2021 - pre-GDB - Worker Nodes

The CERN IT Batch farm in numbers…

…How do we get there?

300K*
Total Cores

1.5M
Jobs Completed / Day

15K
Worker Nodes Virtual & Physical

1.5PB
Total Memory

350
Unique users Daily

2
HTCondor Clusters

> Procurement <

Provisioning

Configuration

Tendering process

Market Surveys

Lab Tests

LHC Req.

Tender Meetings

CPU
HepSpec06 / core

80+ Power Supply

Memory
3 GB / core

2 x 10 Gbit

Disk
2 local SSDs 3-4TB

BMC

Best ratio $/hepsepc?

Technical specifications

CPU: No vendor specific requirements

- The specification is based on total HepSpec06 based on SMT-on cores
- 2 processors per board / reduce infrastructure overhead

Memory: 3GB/core

- Official LHC requirement is 2GB/core
- 3GB/core compensates virtualisation overhead and non-LHC requirements
- Worker nodes and service nodes have same requirements
- Monitor memory prices

Technical specifications

Disk

- 2 local SSDs, enterprise level. Good lifespan.
- Total storage around 3-4TB (OS + extra software + storage/job)

Other details

- 80+ power supply
- Connectivity with 2 x 10 Gbit ports for data and management
- BMC details

> Ensure smooth operations, based on previous experience and tests <

Acceptance

Burn-in

- CPU: burn tools like burnK7, burnP6 and burn MMX
- Memory: memtest
- Disks: badblocks and SMART counters looking for relocated bad blocks

Benchmark

- Obtain HepSpec06 for the hardware
- Measure disk performance with fio and networking with iperf

Tender conditions require a successful burn-in execution

[1] Hardware burn-in in the CERN datacenter

Procurement

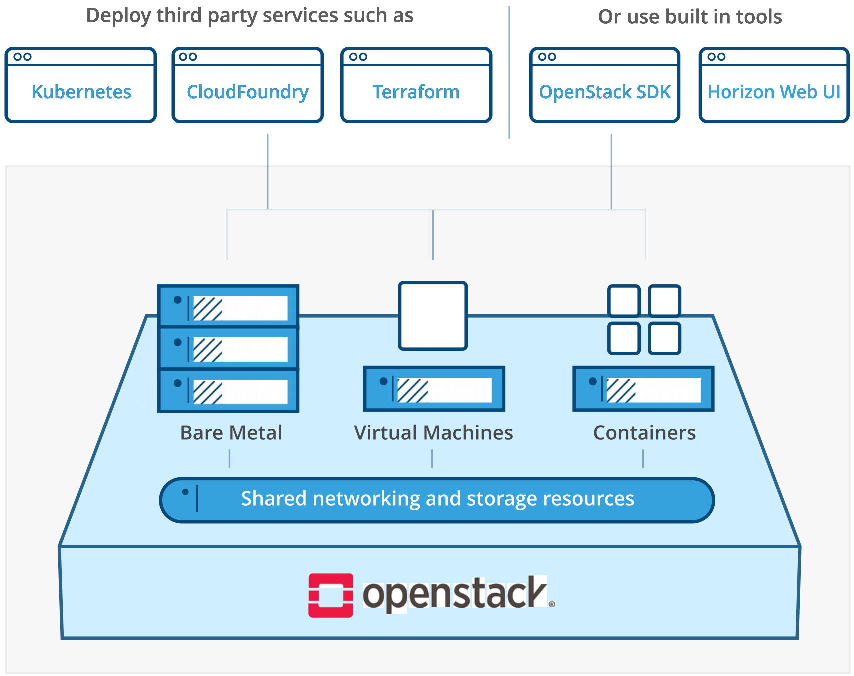
> Provisioning <

Configuration

OpenStack, our interface

Different models

- Vast majority of our farm is based on Virtual Machines
- We are currently moving towards Bare Metal using OpenStack Ironic
- Other models used in our farm: Opportunistic, Pre-emptible and Kubernetes



Hypervisor tweaks for virtual machines

CPU Pinning

- Virtual cores are pinned to physical ones
- It ensures all VM cores are placed in the same NUMA node

Huge Pages enabled

- Reduce overhead by enabling kernel huge pages of 2MB size

CPU Mode

- Hypervisors configure CPU mode as ‘passthrough’
- Expose full CPU capabilities at the cost of live-migration (not essential for worker nodes)

[1] Optimisations of the Compute Resources in the CERN Cloud Service
[2] NUMA and CPU Pinning in High Throughput Computing

Why migrate to Bare Metal?

Get virtualisation tax back

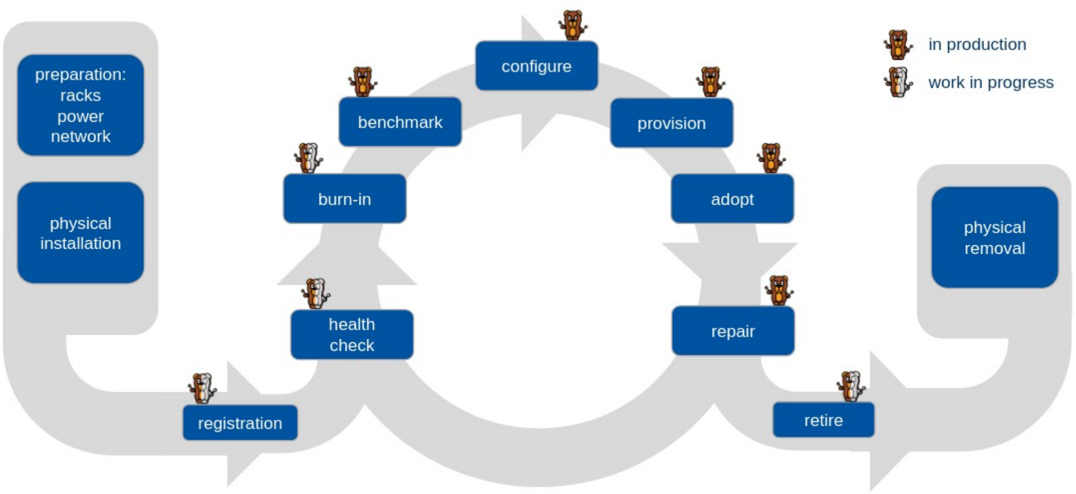
- Estimated on 5%

Same APIs as VMs

- Bare metal nodes provisioned using OpenStack Nova API as we do for VMs.

Re-use existing automation

- Resources provisioned with Terraform



[1] The Case of Ironic in CERN IT

Other provisioning models

Kubernetes based worker nodes

- Exploratory work for fabric management
- Leverage Kubernetes built-in logic for operations
- Based on bare metal nodes
- Some limitations under investigation before moving forward.

Opportunistic (Pre-emptibles)

- Take advantage of unused capacity
- Pre-empted when it is needed by the rightful owner

Opportunistic (BEER)

- Batch on EOS Extra Resources
- Run batch on storage servers (low CPU usage)

[1] Preemptible Instances in production at CERN
[2] Managing the CERN Batch System with Kubernetes
[3] Sharing server nodes for storage and computer

Procurement

Provisioning

> Configuration <

How are the worker nodes configured in our farm?

Configuration

- VMs and bare metal configured with Puppet
- It deploys HTCondor, storage (AFS, EOS, CVMFS), monitoring and base software dependencies

HTCondor

- Worker nodes are added to our HTCondor pools
- Each machine is exposed in a HTCondor partitionable slot
- Two pools, one “shared” and one “T0” pool with dedicated resources

Resource allocation for jobs

SMT-on cores, no overcommit

HTCondor is configured to run jobs on cgroups

- It defines the cpu shares and the memory requested
- No hard memory limit policy and swap enabled
  - Jobs can allocate memory beyond the requested amount if there is no contention
  - If other jobs request memory, jobs exceeding limits are pushed back to the limits

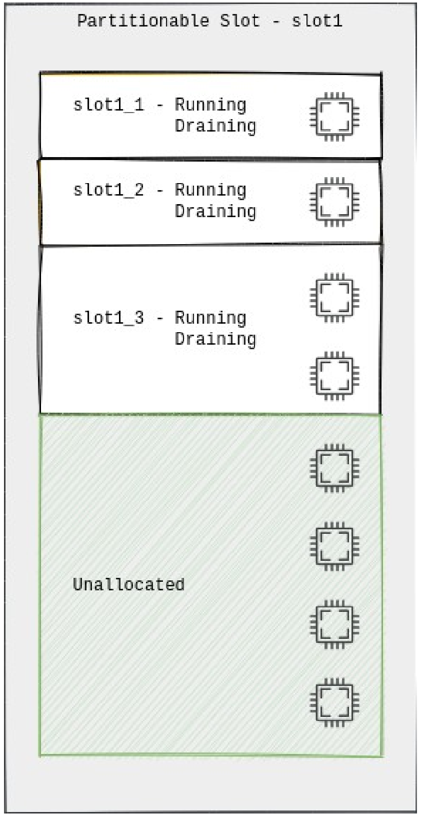
No CPU affinity

- Jobs are allowed to use more than the requested CPUs if no pressure (cgroup shares)

Fragmentation

Our cluster runs a mix of single-core, multi-core (8) and arbitrary job sizes

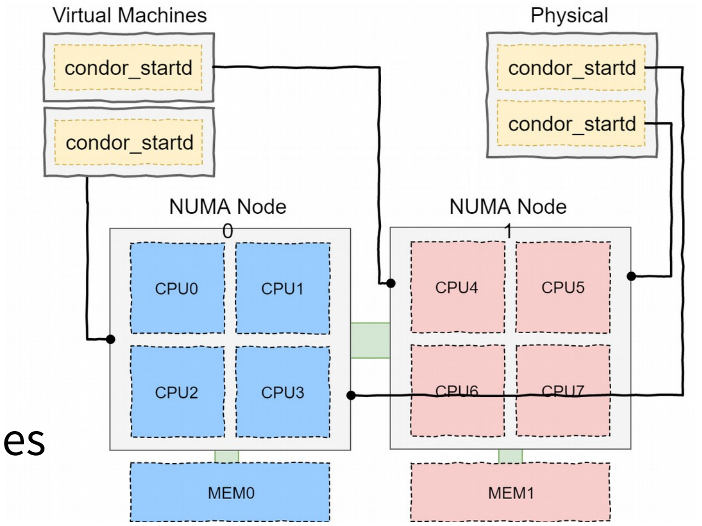
- Fragmentation becomes a problem as the vast majority of the job requests are single-core
- How to find a fair allocation for multi-core jobs?
  - Current approach: condor_defrag. Drain nodes to allocate multi-core jobs.
  - Once a machine has at least 8 cores available, it only accepts multi-core jobs for a few negotiation cycles.
- Challenge to find the sweet spot of concurrent number of machines to defrag



Moving to Bare Metal

Some benefits

- Remove the virtualisation overhead (~5%)
- Single CVMFS cache for same amount of cores
- Similar provisioning and configuration mechanics



Work in progress

- New operation procedures to handle hardware repairs (synergies with existing Cloud procedures)
- Optimise CPU usage with NUMA pinning of condor_startds

> Future-proof ? <

Should we change how we provision, schedule or buy?

Impact of new software on memory requirements?

- Previous discussions with ATLAS about new multithreaded software to be more memory efficient. How does this change the requirements? Do other experiments face similar situation?

Multi-core jobs everywhere?

- Are LHCb and ALICE going to run multi-core jobs?

Is 8 core the standard multi-core size?

- CMS has been using 8 core and “full node” nodes, what’s the best mix?
- ATLAS has run 8 core jobs, will it still be the standard? Might new software impact this?

CERN

home.cern