CERN

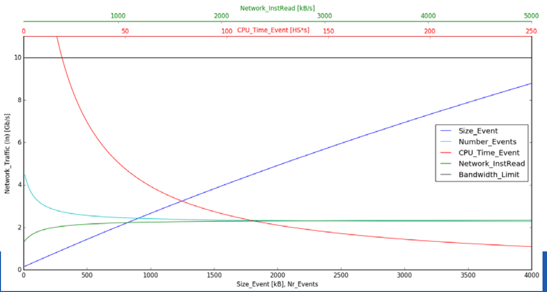
Workflow Performance

Gerhard Rzehorz

Optimizing and Modeling

- Adapt infrastructure to workflow? ⟶ Cloud
- Trade CPU for RAM, Storage for Network
- Dedicated Clusters for specific workflows?
- Storage-less sites, latency hiding

ATLAS Real Data Reconstruction				
Number of Processes	RAM [GB]	Data location	Overall node throughput [s/event]	Overcommit Improvement [%]
8	32	BNL	4,31	Yes:
2x8	32	BNL	2,45	43
8	16	BNL	4,22	Yes:
2x8	16	BNL	3,51	16
8	32	local	3,02	Yes:
2x8	32	local	2,33	22
8	16	local	3,17	No:
2x8	16	local	3,37	-6



CERN Cloud Infrastructure Report

Arne Wiebalck
for the CERN Cloud Team

HEPiX Spring Meeting
DESY, Zeuthen, Germany
Apr 19, 2016

Numbers

Operations

What's new

WIP

The "20% overhead" problem

- On our batch full node VMs we noticed that the HS06 rating was ~20% lower than on the underlying host
  - Full node VMs are needed to the limit of the total number of hosts in LSF

- Smaller VMs behaved much better: ~8%
  - The sum of simultaneous HS06 runs on 4x 8-core VMs on a 32-core host
  - Better, but still pretty high

- IN2P3 reported significant performance penalties for ATLAS MC jobs when EPT* was switched on
  - 26% vs. 6% in wall clock time for EPT on vs. EPT off compared to bare metal
  - Surprising as EPT is supposed to make things faster

***EPT:** Extended Page Tables is Intel's implementation of a hardware-assisted virtualization technology for page table management (secondary address translation or nested pages). AMD's implementation is called RVI (Rapid Virtualization Indexing).

The “Kilo-1” configuration

- NUMA + Pinning
  - 1-to-1 vs. 1-to-N no difference
- 2MB huge pages
  - 1GB slightly better
- EPT on
  - EPT off still better in HS06

VM sizes (cores)	Before	After
4x 8	7.8%	3.3% (batch WN)
2x 16	16%	4.6% (batch WN)
1x 24	20%	5.0% (batch WN)
1x 32	20.4%	3-6% (bare SLC6 … batch WN)

ATLAS T0 host with batch VM running the new config: throughput for recon jobs 20% higher!

OpenStack Kilo will fully support our desired configuration!

Integrating Commercial Clouds into the WLCG

Cristovao Cordeiro

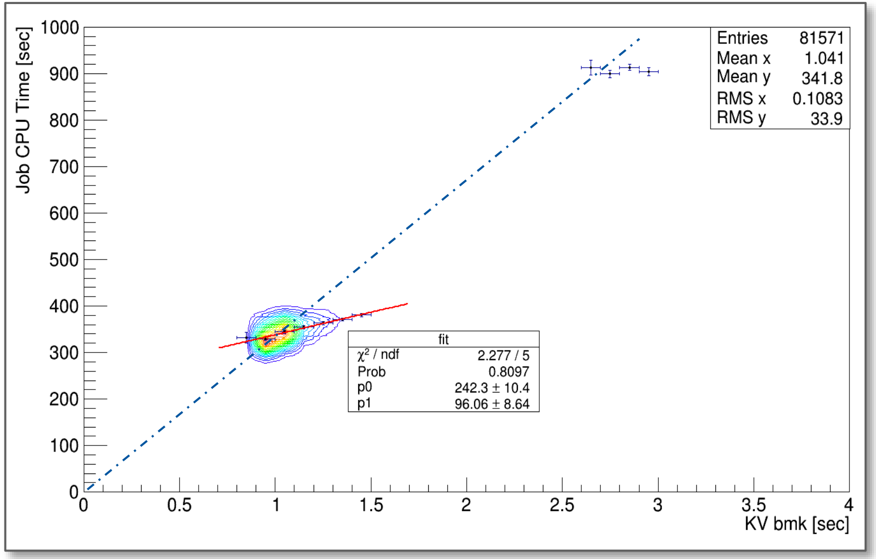
on behalf of the WLCG Resource Integration team

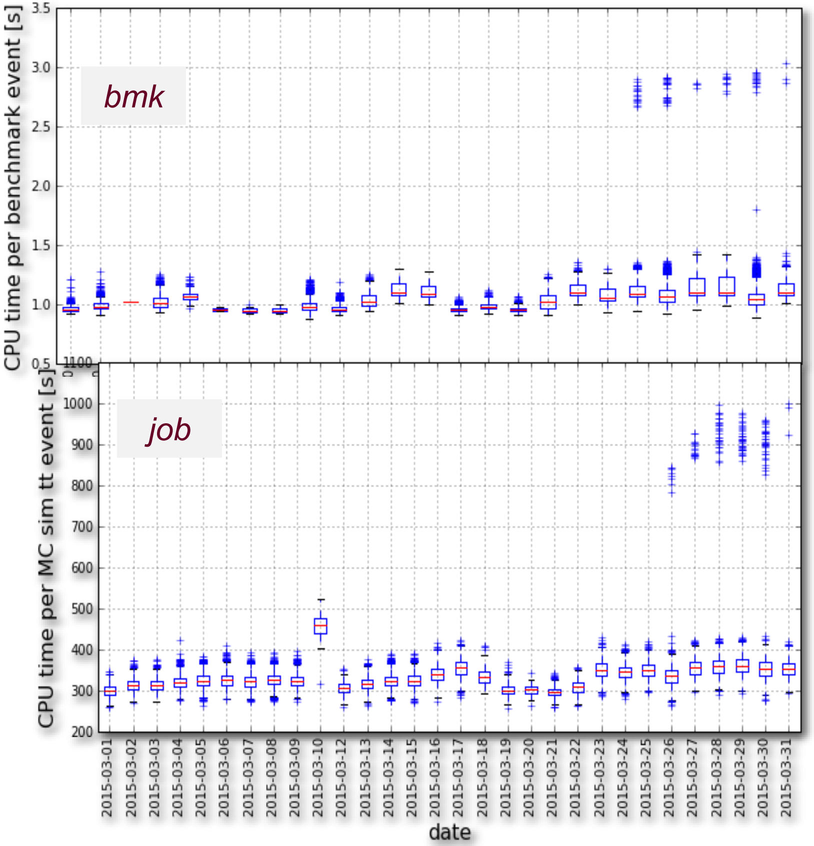
IT Technical Forum, CERN, 22 January 2016

Deploy & *monitor* & *account* & *benchmark*

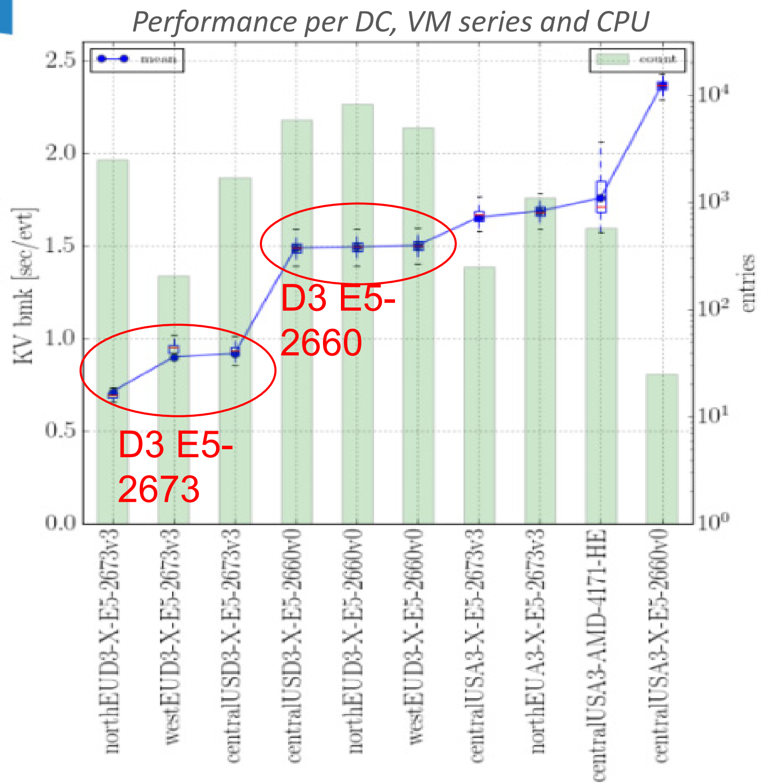
Atos

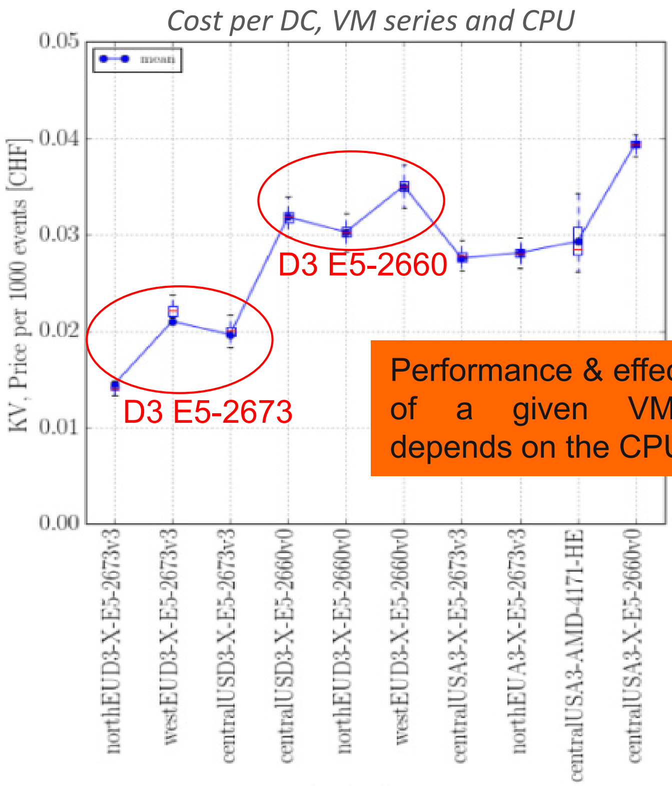
- Correlated job CPU performance and benchmark
- Results from ~30k VM benchmarks, simulating 100 Single Muon events each (~2' to run)





Deploy & *monitor & account & benchmark*



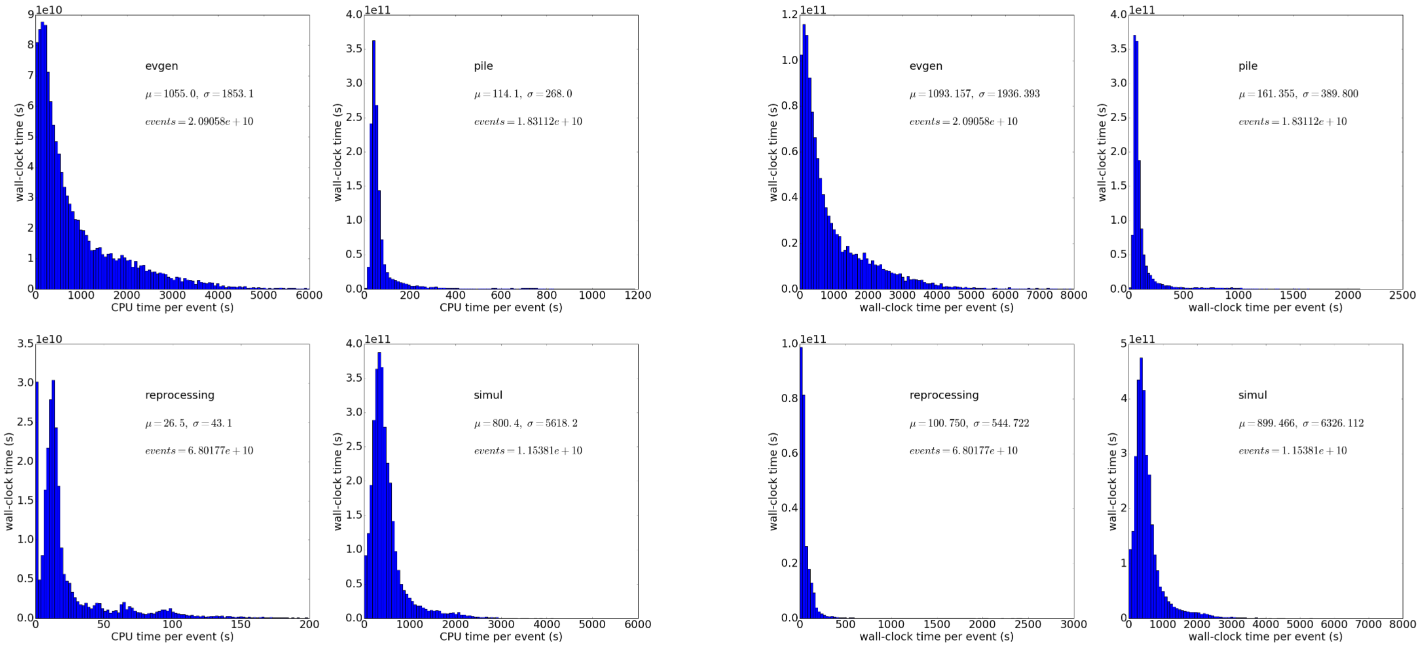


Performance & effective cost of a given VM series depends on the CPU model

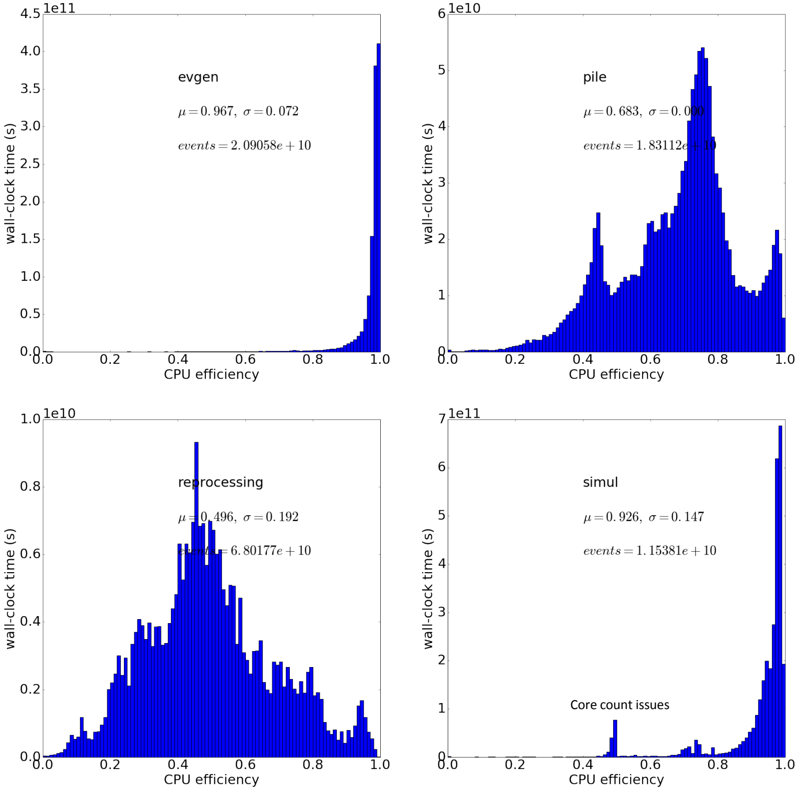
CPU utilization of production jobs

Andrea Sciaba

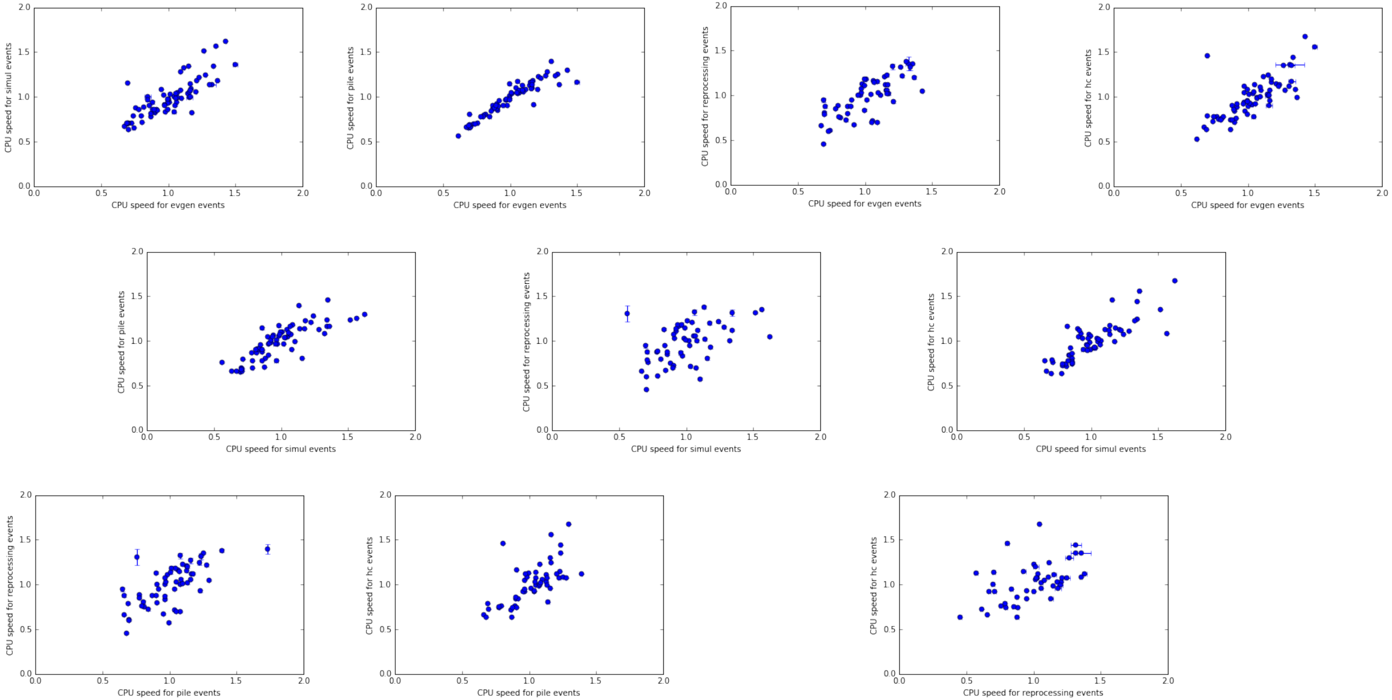
ATLAS CPU and wall-clock per event averages for tasks



ATLAS CPU efficiency averages for tasks

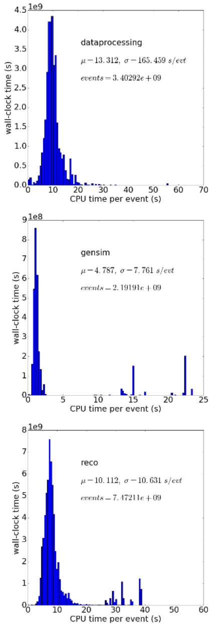


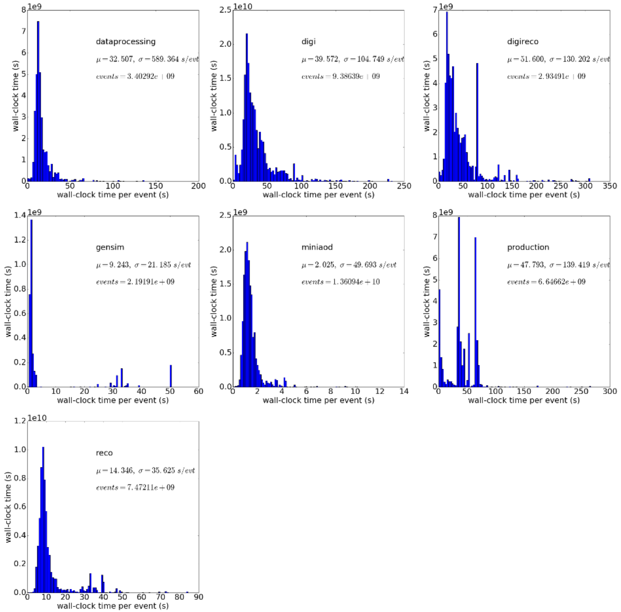
Speed correlations



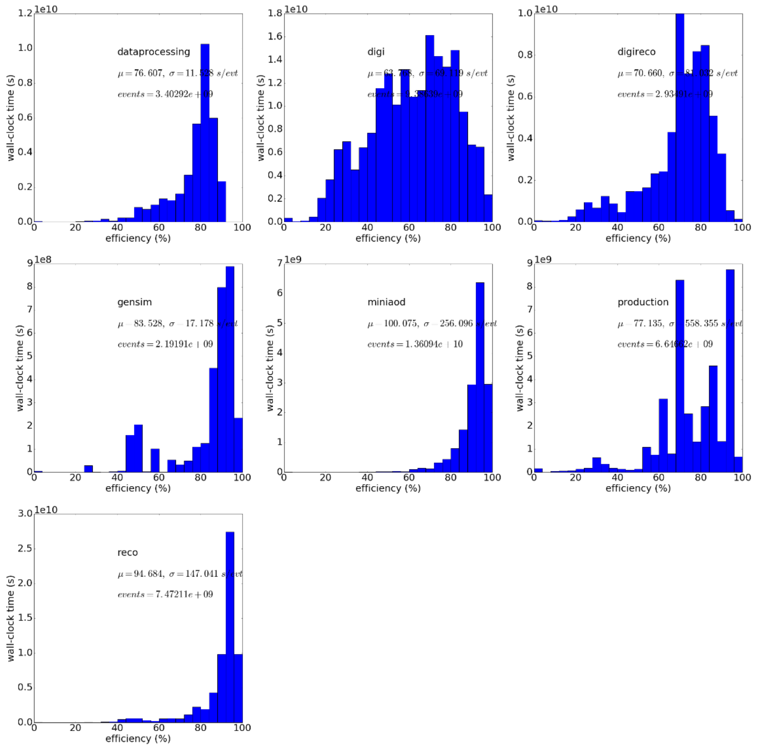
When considering only the three main categories (evgen, simul, pile), the speed factors agree by ~4% on average

CMS analysis

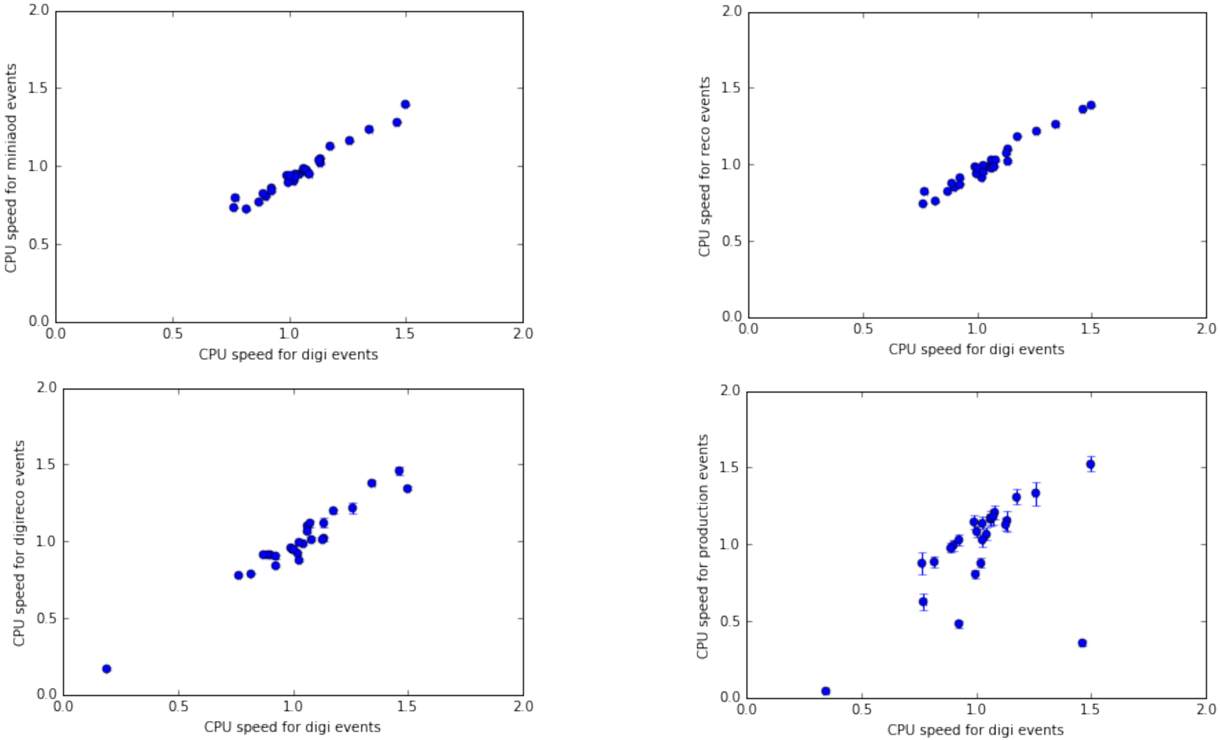




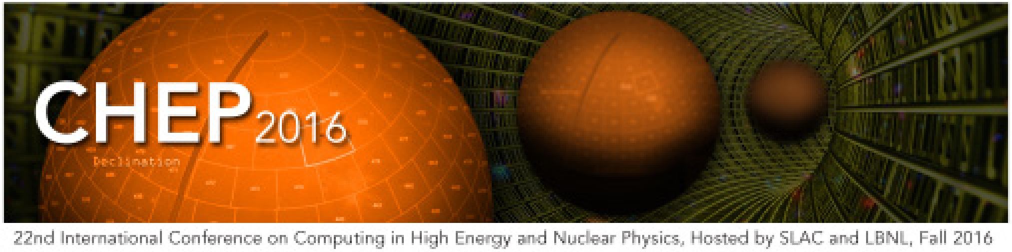
CMS CPU efficiency



Speed correlations in CMS



When considering only digi, miniaod, digireco and reco, the speed factors agree by ~4% on average

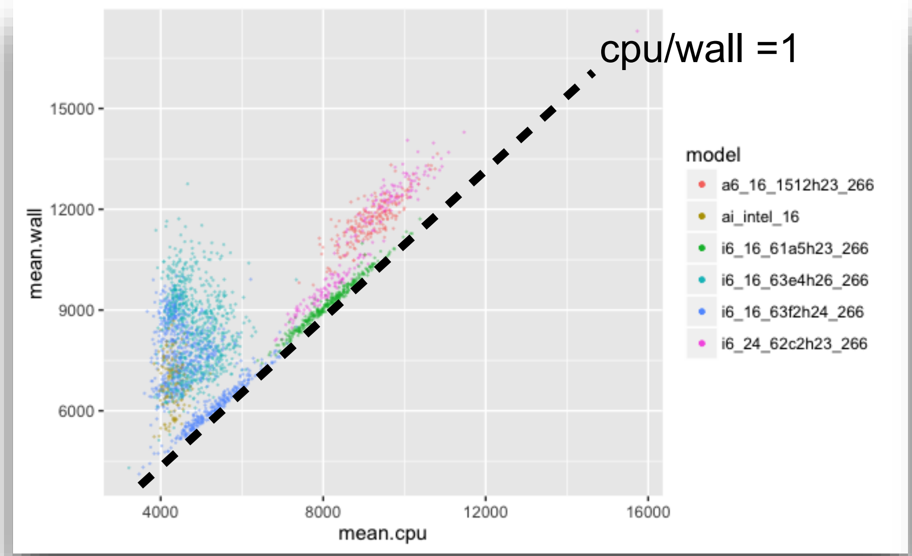


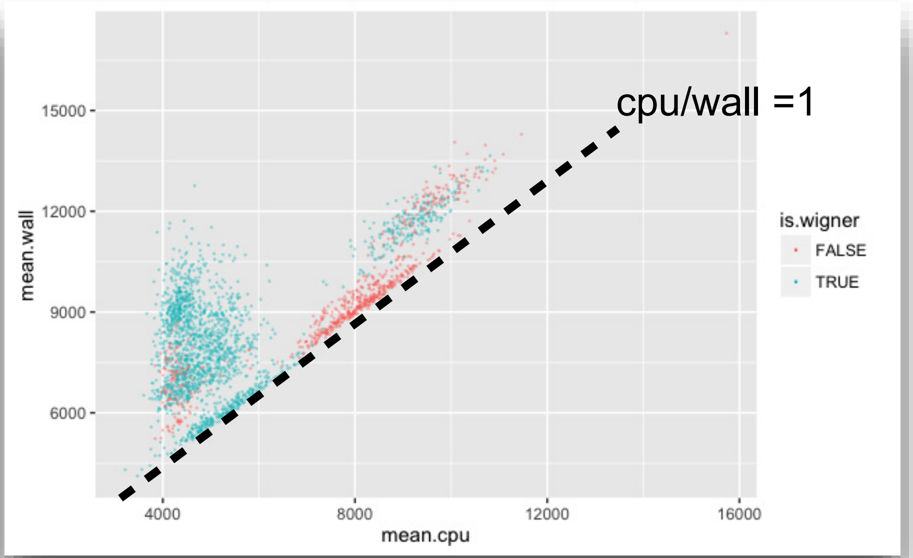
First results from a combined analysis of CERN computing infrastructure metrics

D. Duellmann - CERN, Switzerland
C. Nieke - Tech. Univ. Braunschweig, Germany

presented by Oliver Keeble

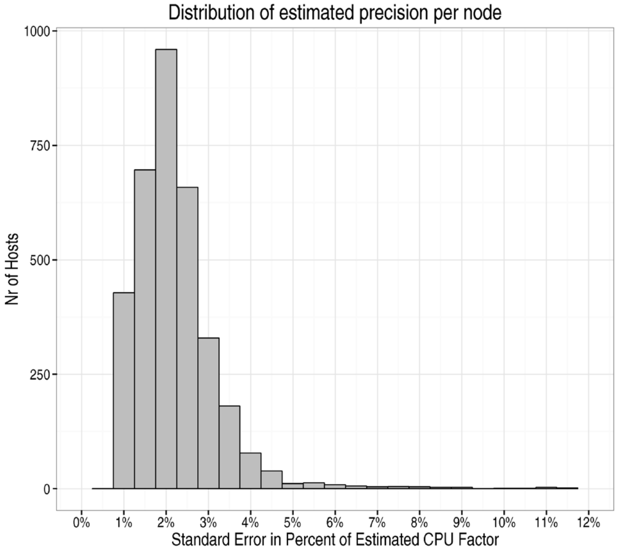
A few simple examples - machine "efficiency" versus H/W and location



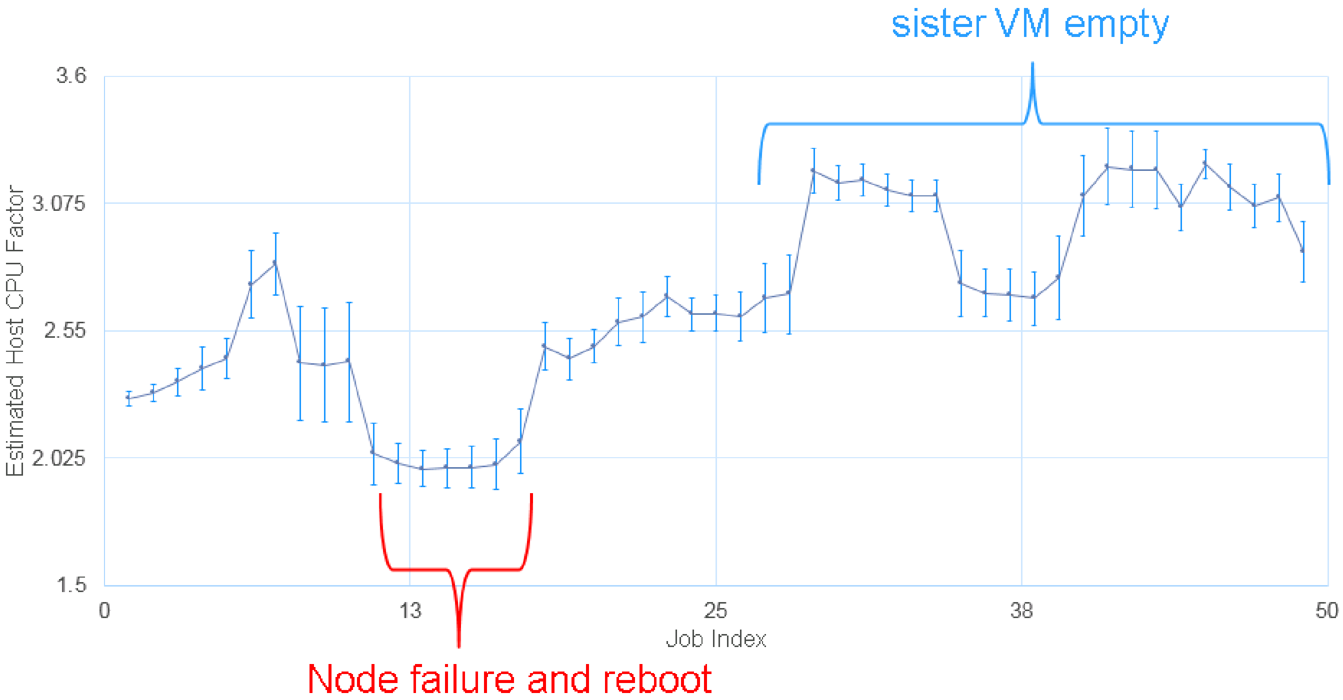


Passive Benchmark

- Basic Idea:
  - Take the workload as set of benchmarks
    - Assume jobs per task are equal, compare runtime
    - Based on existing monitoring logs
- Advantages:
  - Zero intrusion, basically no overhead
  - Always representative (the benchmark **is** the workload)
- Application:
  - Observe performance during operation
  - Compare configurations by performance on the actual workload
- Accuracy / Precision
  - Experiment on LSF dataset: ATLAS and CMS, 3 months
  - Equal or better prediction of performance than HepSPEC06
  - Precision per node is below 5% error for 98% of nodes



"Live" Performance Monitoring



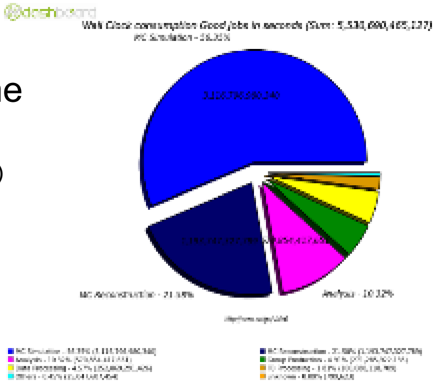
- CPU performance computed on sliding window of 4 tasks

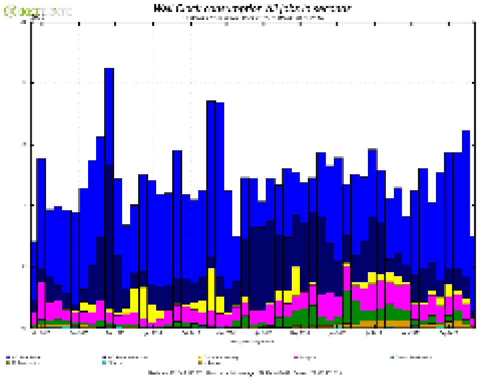
Backup

Example: ATLAS jobs in the last year

By wall-clock time

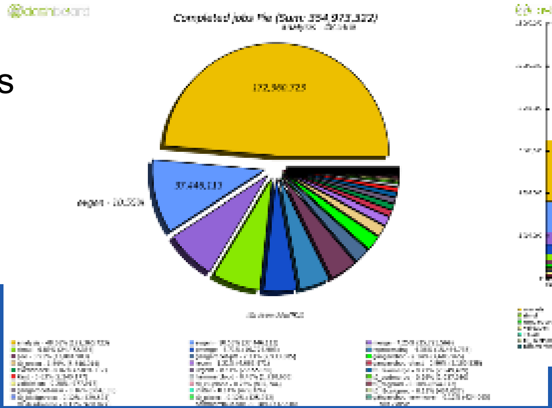
- MC Simulation: ≈ 60% (of which 2/3 simul, 1/3 evgen)
- MC Reconstruction: ≈ 22% (of which 4/5 pile)
- Analysis: ≈ 10%
- Data processing: ≈ 5% (mainly reprocessing)
- Group production: ≈ 5% (mainly merge)

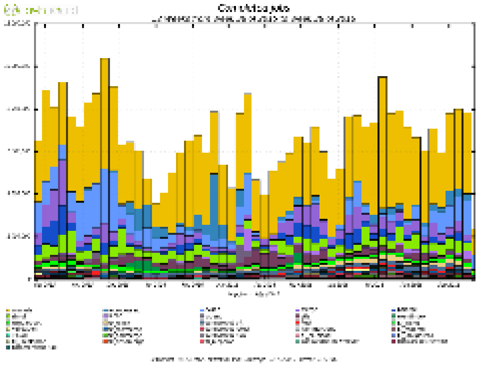




By number of jobs

- Analysis: ≈ 50%
- Evgen: ≈ 10%
- Merge: ≈ 7%
- Simul: ≈ 7%
- Pmerge: ≈ 5%
- Reprocessing: ≈ 4%
- Pile: ≈ 4%





General virtualization issue?

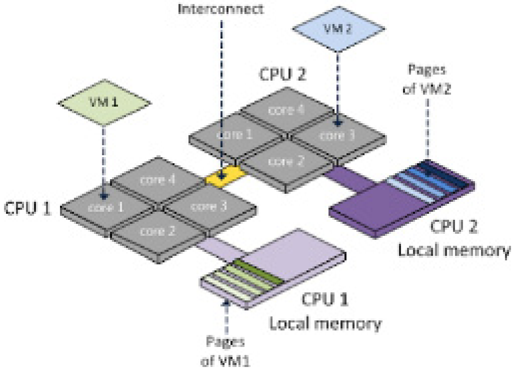
- Crosscheck w/ SLC6 VMs on Hyper-V
  - 0.8% HS06 loss on 4x 8-core
  - 3.3% HS06 loss on 1x 32-core SLC6 VM

Microsoft Hyper-V

- No general virtualization overhead issue!
  - Rather a feature or configuration issue

- What’s the difference between the VMs on Hyper-V and the ones on KVM?

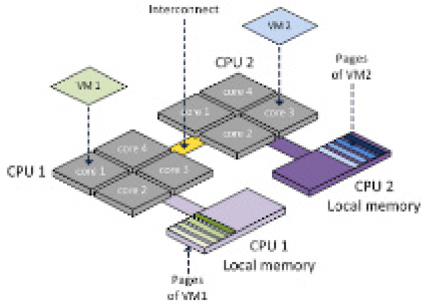
NUMA



- Hyper-V VMs “see” underlying NUMA architecture
- Hyper-V VMs have vCPUs pinned to physical NUMA nodes
  - Pinned to sets that correspond to physical NUMA nodes
- In OpenStack, wider support for this comes with the release we currently prepare to deploy (Kilo)
  - Juno has some NUMA support as well
  - Consciously not configured, then disabled after ATLAS T0 incident (aka the “50% overhead” problem …)

Operations: NUMA/THP Recap (2)

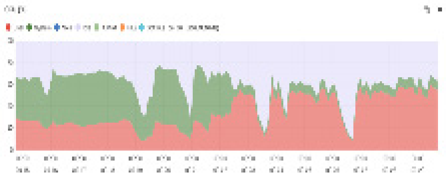
- NUMA-awareness identified as most efficient setting
  - Full node VMs have ~3% overhead in HS06
- “EPT-off” side-effect
  - Small number of hosts, but very visible there
- Use 2MB Huge Pages
  - Keep the “EPT off” performance gain with “EPT on”
- All details in Arne Wiebalck’s talk at BNL



Operations: NUMA/THP Roll-out

- Rolled out on ~2’000 batch hypervisors (~6’000 VMs)
  - HP allocation as boot parameter → reboot
  - VM NUMA awareness as flavor metadata → delete/recreate

- Cell-by-cell (~200 hosts):
  - Queue-reshuffle to minimize resource impact
  - Draining & deletion of batch VMs
  - Hypervisor reconfiguration (Puppet) & reboot
  - Recreation of batch VMs



- Whole update took about 8 weeks
  - Organized between batch and cloud teams
  - No performance issue observed since

Deploy & *monitor & account & benchmark*

> “*Performance is a key criterion in the design, procurement, and use of computer systems […] to get the highest performance for a given cost.*” [1]

- Together with monitoring and accounting, Performance Measurement is also essential
  - Deal with the intrinsic variability and inhomogeneity
  - Compare the presumed and perceived performance
  - Identify performance issues

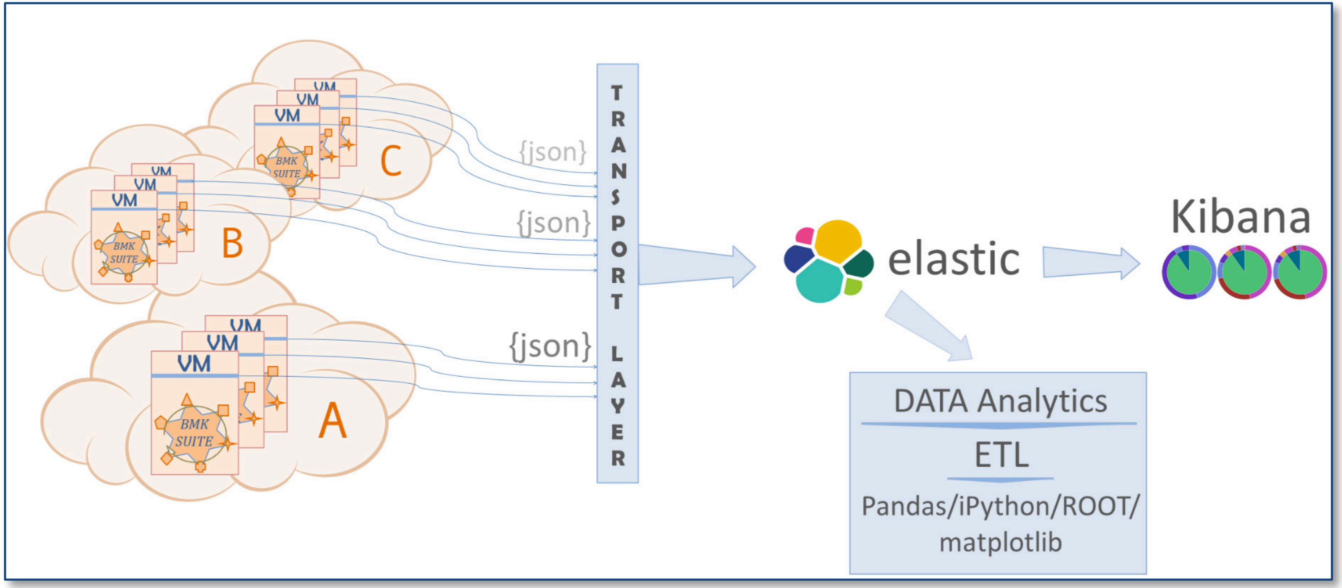
- Standard procedure during the procurement process
- Built a Cloud Benchmarking Suite that runs on every cloud resource and currently includes:
  - ATLAS Kit Validation (KV)
  - LHCb Fast Benchmark (fastBmk)
  - 4 different open-source Phoronix tools

Focus on CPU performance metric ➔ CPU time/event

[1] *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling* by Raj Jain , Wiley Computer Publishing, John Wiley & Sons, Inc

Deploy & *monitor & account & benchmark*

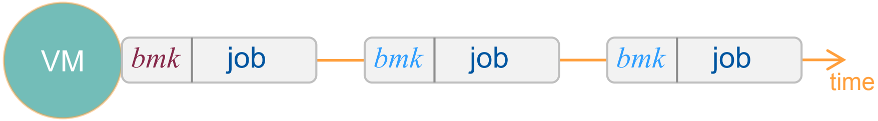
- A scalable architecture whereby each benchmarking suite and respective configurations are deployed independently, on each cloud resource, through the contextualization mechanism



- All results are inserted into ElasticSearch, containing unique metadata information on a VM level

Deploy & *monitor* & *account* & ***benchmark***

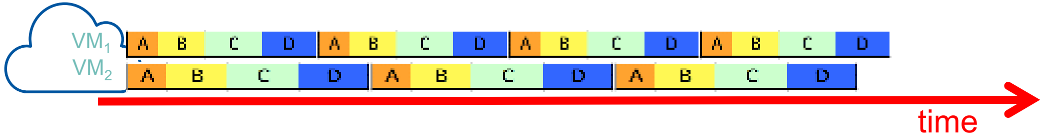
- Run the benchmark in parallel in many threads as the number of vCPUs
- The benchmark suite can be run at any time during the VM lifecycle
  - Only at the *beginning*
  - At *each job cycle* (bmk included in the job wrapper)



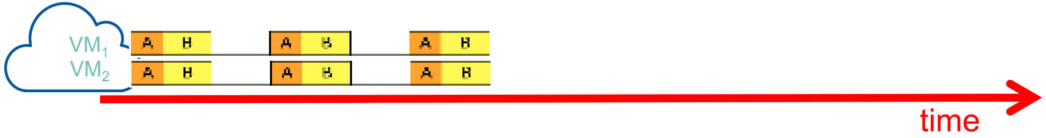
PRODUCTION

STRESS TEST

  - Sequentially - all configured benchmarks in a row

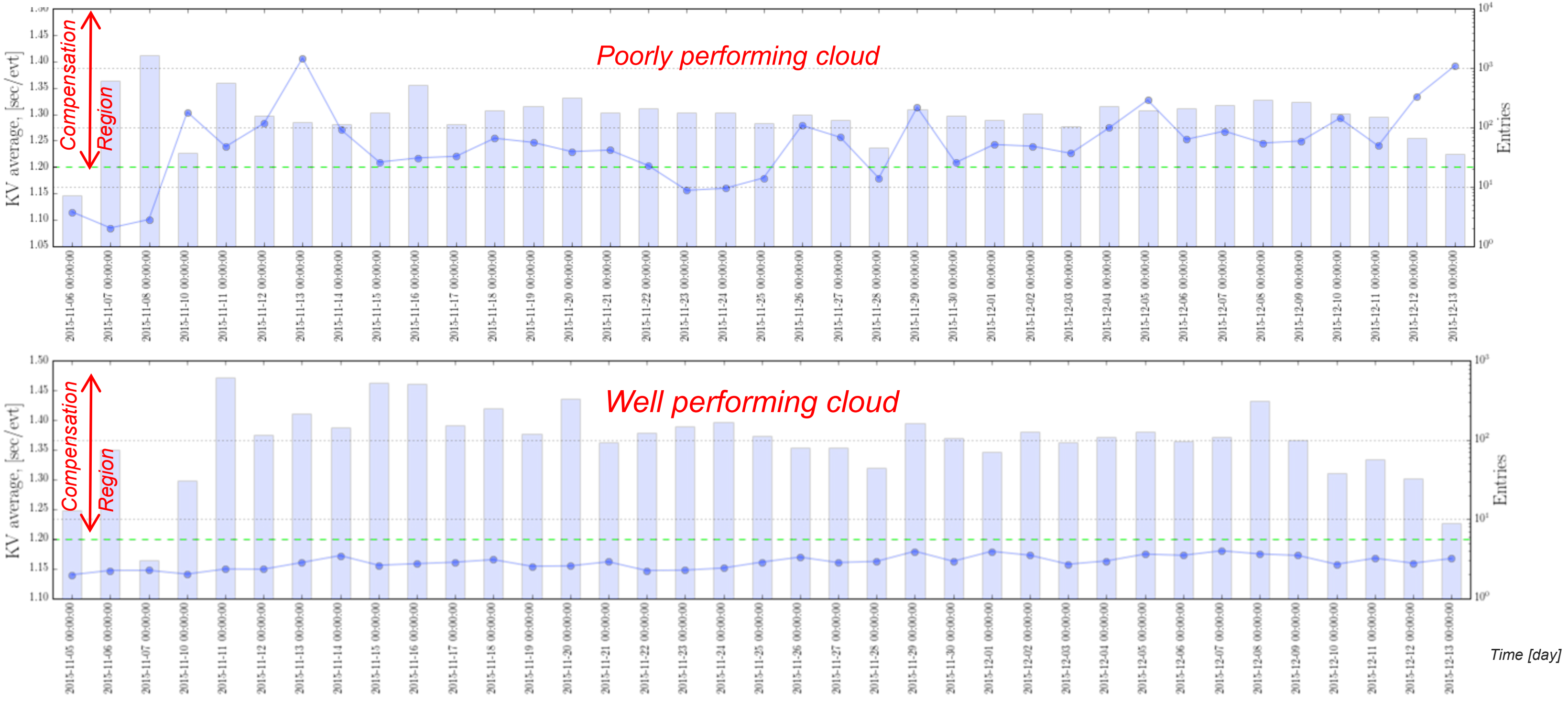


  - Synchronized - specific benchmarks at specific points in time



Deploy & *monitor* & *account* & *benchmark*

DEUTSCHE BÖRSE
CLOUD EXCHANGE



- **Bad performances might trigger contractual reviews and compensation measures**

Motivation

- Understand how experiments use their CPU resources
  - What types of jobs are (primarily) run?
  - How many resources do they require?
  - Are they “efficient” (i.e. do they waste wall-clock time)?
  - How can jobs be modelled in the context of a simulation of the WLCG computing infrastructure?
- Using data analytics techniques, understand the behavior of the infrastructure
  - Can we measure the “speed” of CPUs, or sites, by looking at different types of jobs? Are the results compatible? Can we validate commonly used benchmarks using “real” jobs?

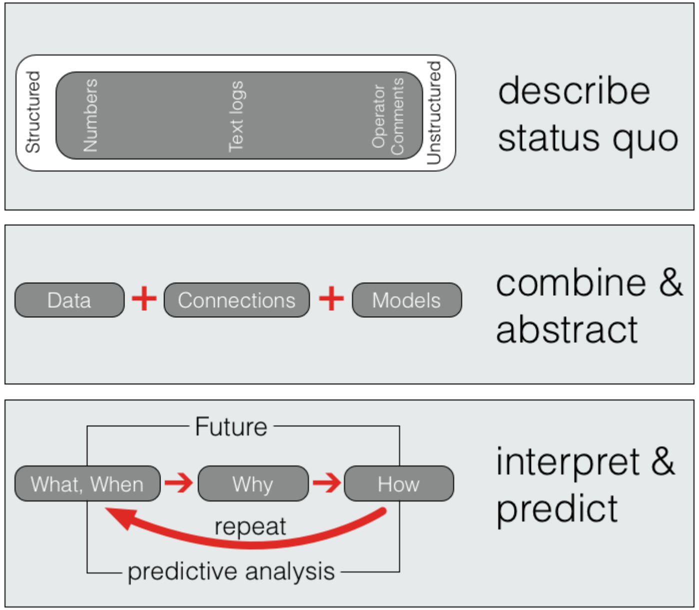
Fitting CPU speeds

- Based on the assumption that the CPU speed and the average CPU time / event are inversely proportional
- Exploiting the fact that jobs in a task are comparable, relative speeds of CPUs they run on can be measured
- Global least-squares fit produces separate CPU speed measurements for different types of jobs
  - Tried on ATLAS jobs for evgen, simul, pile, reprocessing and selected HammerCloud jobs

Analysis Working Group Motivation

- Goal: **quantitative** understanding of computing infrastructure
  - measure service usage and their evolution
  - To optimise science throughput / CHF
- End-to-end: using metrics from user and across layered services
  - Eg within large (many jobs), well-identified physics tasks
    - optimise aggregate events/s delivered to experiments

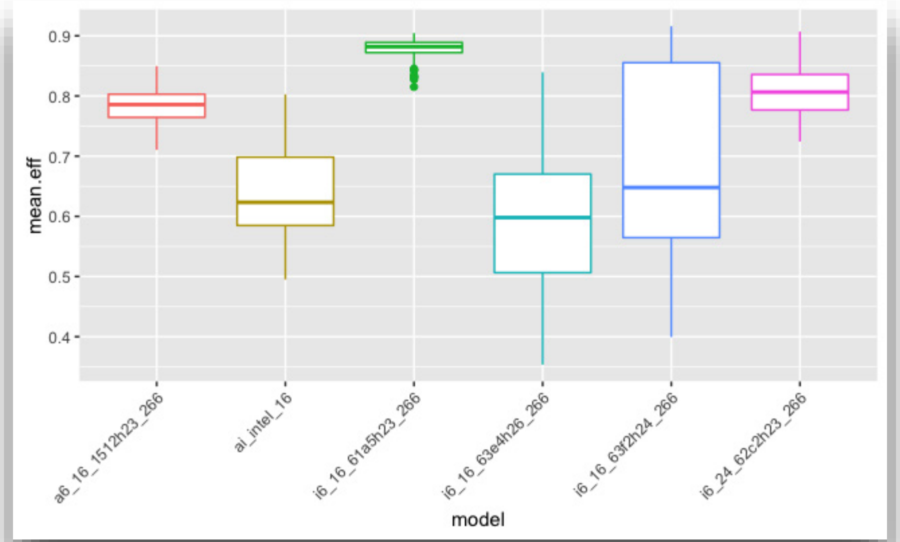
Three Main Phases



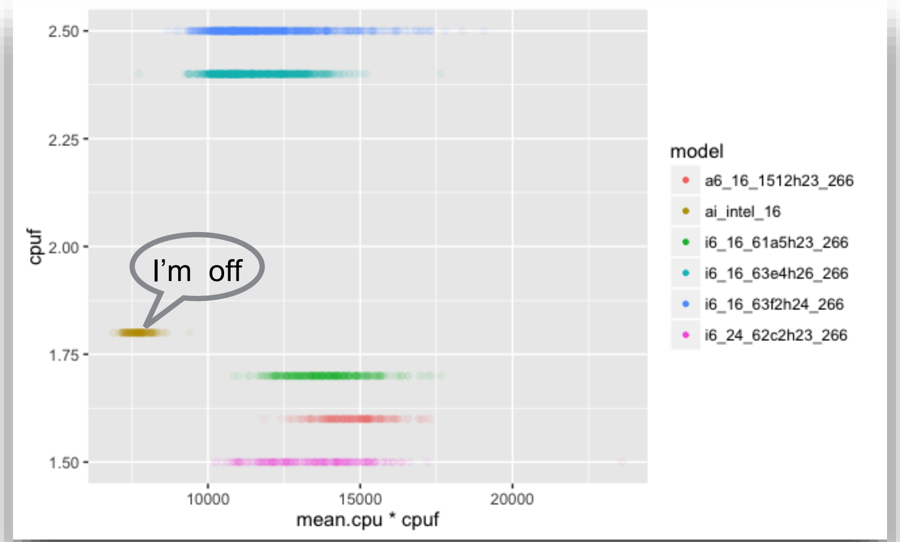
- Tools
  - statistical analysis {machine learning}
- Follow rational flow
  - extract relevant statistics from raw metrics and logs
  - connect trusted data into models
  - check model prediction and improve

A few simple examples

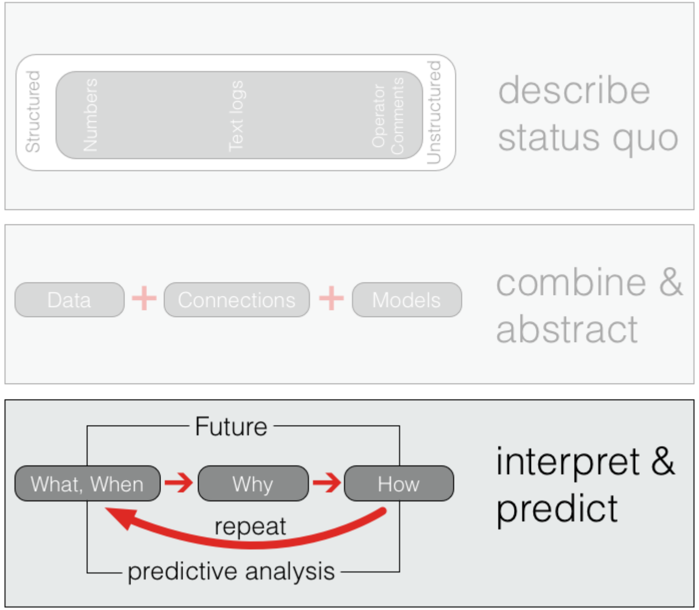
“Efficiency” versus H/W types



CPU Performance Calibration check



Model Predictions



- Evaluate (simple) predictive models: Can we construct a more performant system for the same price?
- Simplest case: CPU bound
  - **CPU benchmarks** + memory optimisation => MC throughput
- Not CPU-bound case - balance between
  - CPU, WN storage, LAN storage, WAN storage, network

CPU Benchmarking Motivations

- Compare hardware performance
  - Before purchasing hardware
- Optimizing software / configuration
  - Maximize performance for a given workflow mix
- Monitor performance
  - Observe shifts over time
  - Detect outliers / failures / bottlenecks
  - Measure current performance of volatile resources
    - (e.g. cloud VMs)

Current Benchmarking Practice

- Long benchmarks (HepSPEC06)
  - High precision (reproducible per machine)
  - Expensive (~8 hours runtime)
  - Intrusive (Node has to be taken out of production)
  - Suited for initial hardware tests
- Short benchmarks
  - Run before each job (e.g. DIRAC, KV, fastBmk etc.)
  - Lower intrusion (10-15 min benchmark vs. ~hours jobs)
  - Suited for hardware monitoring
- But do they represent the actual workflows?
  - HepSPEC designed to represent a mix of HEP workflows
    - But does it?
  - Short benchmarks even more questionable
    - Typically cover only specific areas (e.g. only Monte Carlo)

Summary

- Statistical analysis activity of CERN infrastructure performance has started
  - IT Analysis Working Group [meetings] and [twiki]
- Input metrics and analysis environment are now in place and allowed first quantitative studies
  - close collaboration with IT monitoring project and Hadoop service
- Promising new approach to determine CPU performance directly from the workflow
  - sufficient resolution to spot time dependent efficiency changes
- Growing interest from experiments and other sites in statistical and ML optimisation