# Probing the grid job slot performance with HEPScore

Natalia Szczepanek
Ewoud Ketele, Domenico Giordano
(CERN/IT-GOV-ENG)

on behalf of the HEPiX Benchmarking Working Group

GDB, 10th January 2024

# Main Objective

- Study grid job slot performance in correlation with server resources (load, memory usage, power consumption, etc.)
- Measure performance in units of HEPScore
- Leverage the enhanced HEP Benchmark Suite

# With the support and in collaboration with

ATLAS EXPERIMENT

- Ivan Glushkov,
- Alessandro Di Girolamo,
- Alexander Lory,
- Jaroslava Schovancova,
- Ilija Vukotic,
- Cesare Delle Fratte,
- Andrej Filipcic

LHCb

- Federico Stagni

# HEPScore23 Configuration

- 7 workloads from 5 experiments
- All workloads have the most recent version of the experiment's software:
    - Support x86_64 and aarch64

| Experiment | WL | x86_64/aarch64 |
|---|---|---|
| ALICE | digi-reco | ✅ |
| ATLAS | gen_sherpa | ✅ |
| | reco_mt | ✅ |
| Belle2 | gen-sim-reco | ✅ |
| CMS | gen-sim | ✅ |
| | reco | ✅ |
| LHCb | sim | ✅ |

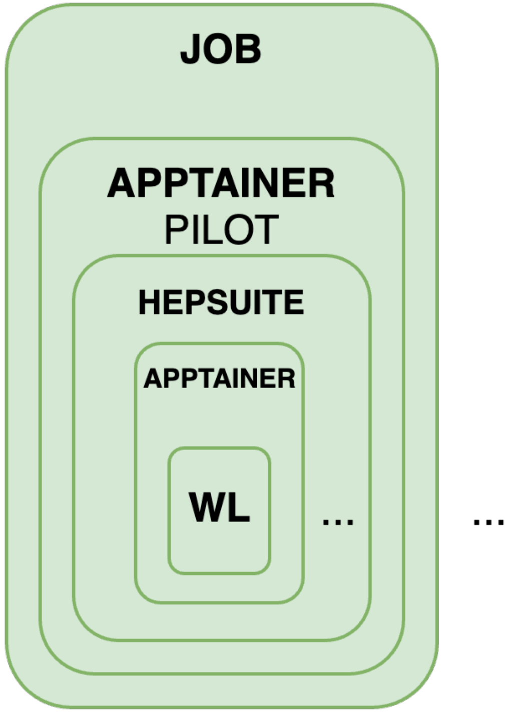
*Doc "How to run HS23": https://w3.hepix.org/benchmarking/how_to_run_HS23.html*

*Benchmarking Working Group: https://w3.hepix.org/benchmarking.html*

*Github: https://gitlab.cern.ch/hep-benchmarks/hep-score*

# Job Slot

- Each site server has a variety of CPU models and number of cores (256, 128, 64…)
- We are running the benchmark injecting the HEP Suite script as a normal experiment job running inside the PILOT Apptainer
- We probe multi-core job slots (8 and 4 cores) and single core job slots
- HEPScore is being calculated at the end as a geometric mean of all WLs if all workloads succeed



JOB

APPTAINER PILOT

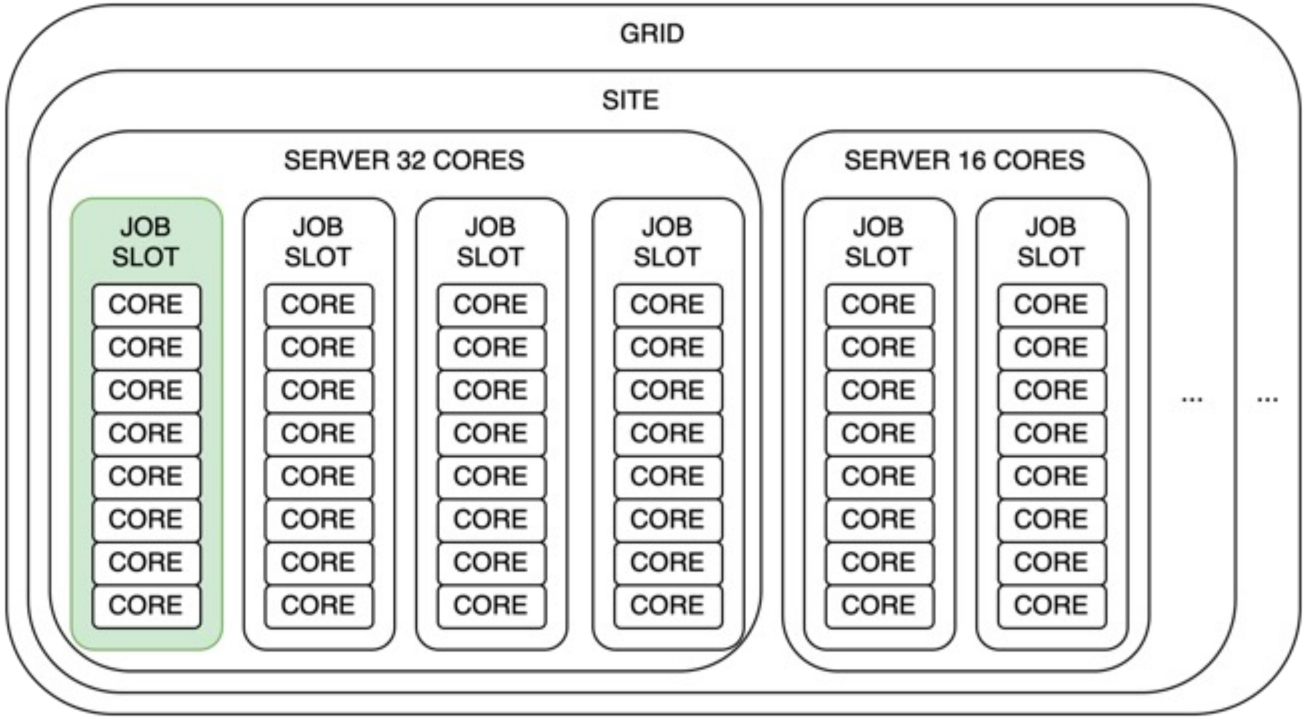
HEPSUITE

APPTAINER

WL

…

…



GRID

SITE

SERVER 32 CORES

JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

SERVER 16 CORES

JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

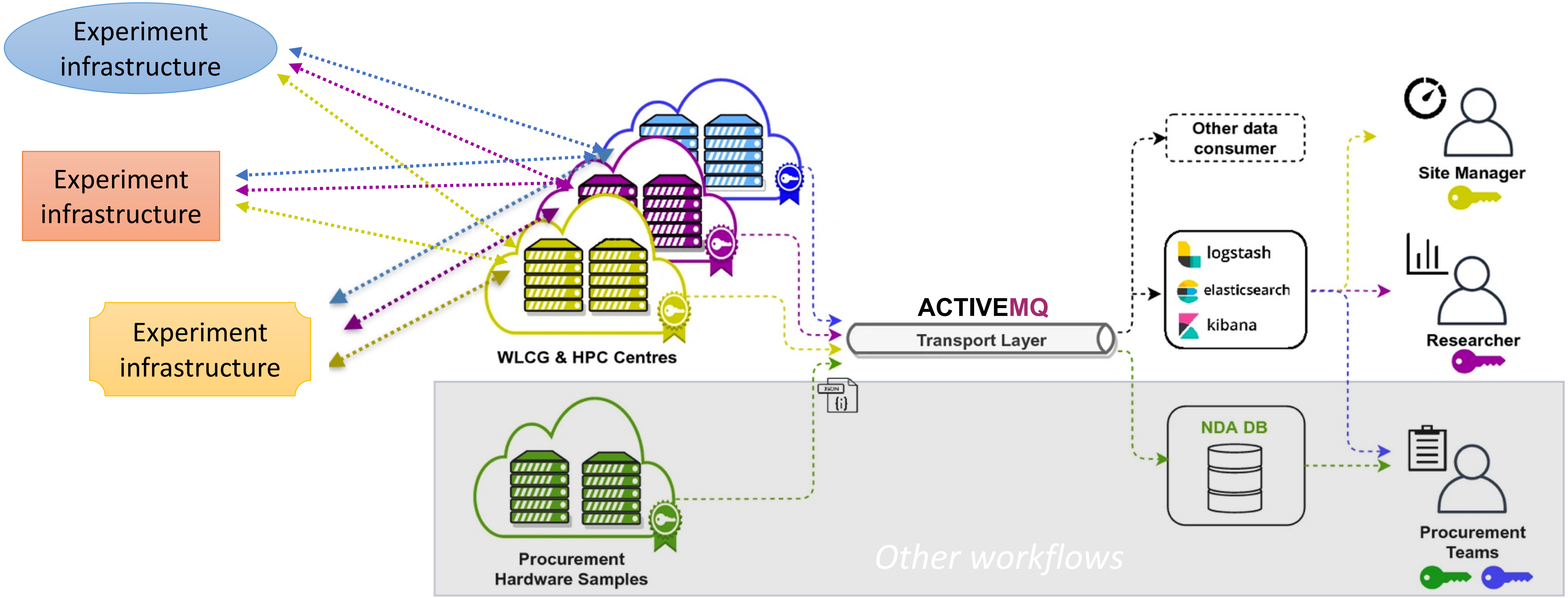
JOB SLOT: CORE, CORE, CORE, CORE, CORE, CORE, CORE, CORE

…

…

# Job Submission

- The Benchmarking Script is injected in the site job slot via the standard job submission system

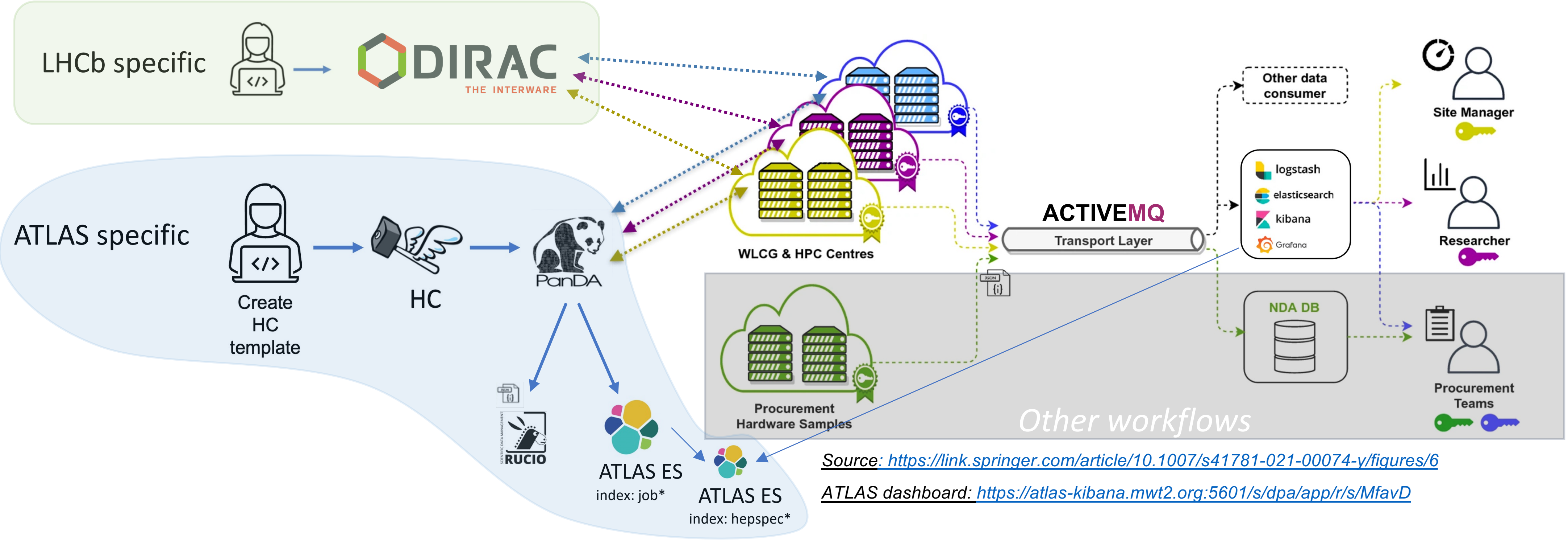


Source: https://link.springer.com/article/10.1007/s41781-021-00074-y/figures/6

# In practice

Successfully implemented and deployed the pipeline for ATLAS and LHCb

- **ATLAS:**
  - Automated submission via HammerCloud
  - Uses: PanDA, Rucio, ActiveMQ, Elasticsearch, Grafana, Kibana…
- **LHCb:**
  - Manual submission to DIRAC



Source: https://link.springer.com/article/10.1007/s41781-021-00074-y/figures/6

ATLAS dashboard: https://atlas-kibana.mwt2.org:5601/s/dpa/app/r/s/MfavD

# ATLAS

data from: 07/04/23 – 08/01/24

- Automated job submission every 3 hours on each panda resource
    - 139 Panda Resources
    - 227 CPU Models
    - 28246 unique hosts
- Over 190k jobs finished
- Each job: 8 core slot
- Median of job's walltime: 81minutes
    - HEPScore23 configuration with 1 repetition
    - 0.06% of total walltime_x_core

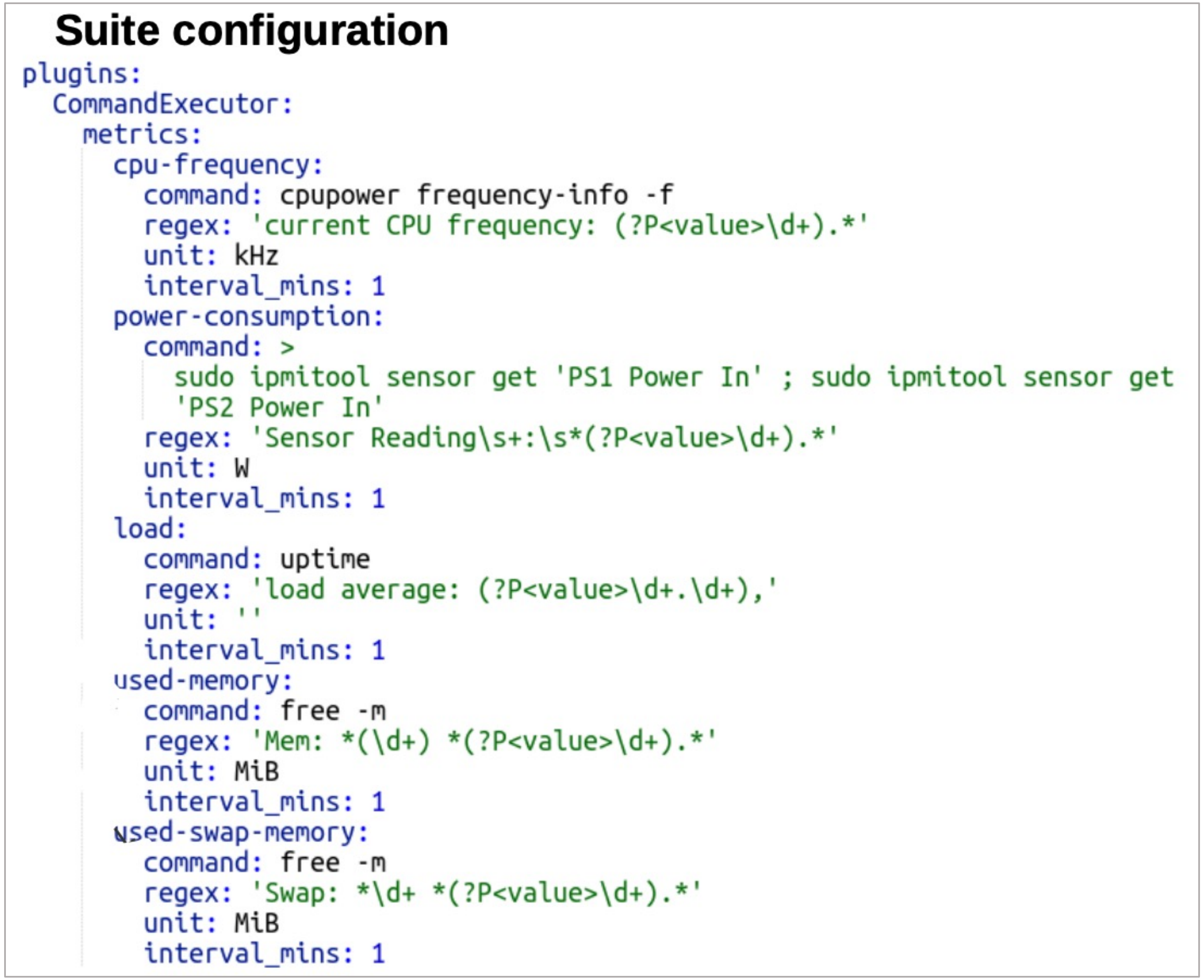
# LHCb

data from: 01/08/23 – 01/11/23

- Manual job submission
    - 48 Sites
    - 110 CPU Models
    - 1650 unique hosts
- 2.1k jobs finished
- Each job: 1 or 4 core slot (most 1core)
- Median of job's walltime: 43minutes
    - lhcb-sim-run3-ma-bmk with 3 repetitions

# Benchmarking with Suite Plugins

- Configurable collection of various system metrics (e.g. load, memory usage, power consumption)
  - Run alongside benchmarks
  - Flexible modification and addition of collected metrics
- Collection configurable at the level of exec command, regex, unit, interval, mins, aggregation

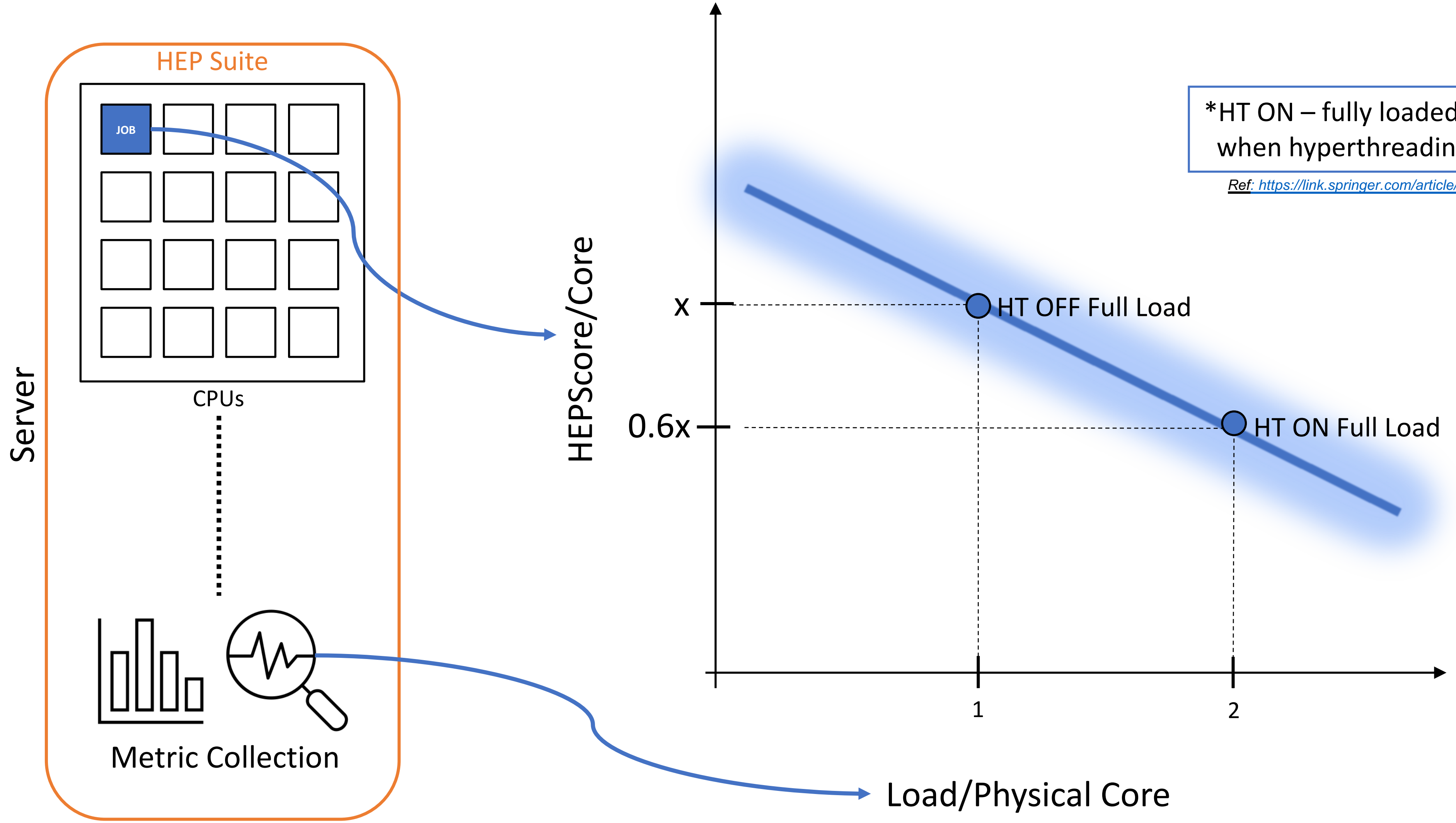


**Suite configuration**

```
plugins:
  CommandExecutor:
    metrics:
      cpu-frequency:
        command: cpupower frequency-info -f
        regex: 'current CPU frequency: (?P<value>\d+).*'
        unit: kHz
        interval_mins: 1
      power-consumption:
        command: >
          sudo ipmitool sensor get 'PS1 Power In' ; sudo ipmitool sensor get
          'PS2 Power In'
        regex: 'Sensor Reading\s+:\s*(?P<value>\d+).*'
        unit: W
        interval_mins: 1
      load:
        command: uptime
        regex: 'load average: (?P<value>\d+.\d+),'
        unit: ''
        interval_mins: 1
      used-memory:
        command: free -m
        regex: 'Mem: *(\d+) *(?P<value>\d+).*'
        unit: MiB
        interval_mins: 1
      used-swap-memory:
        command: free -m
        regex: 'Swap: *\d+ *(?P<value>\d+).*'
        unit: MiB
        interval_mins: 1
```

# Data Analysis

# Performance vs Load Data Model



*HT ON – fully loaded, but not overloaded, when hyperthreading is enabled on x86

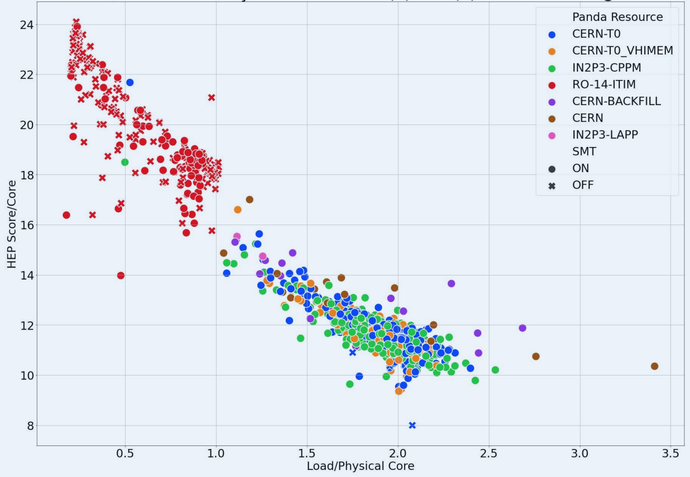
Ref: https://link.springer.com/article/10.1007/s41781-021-00074-y/figures/10

# From the model to real data

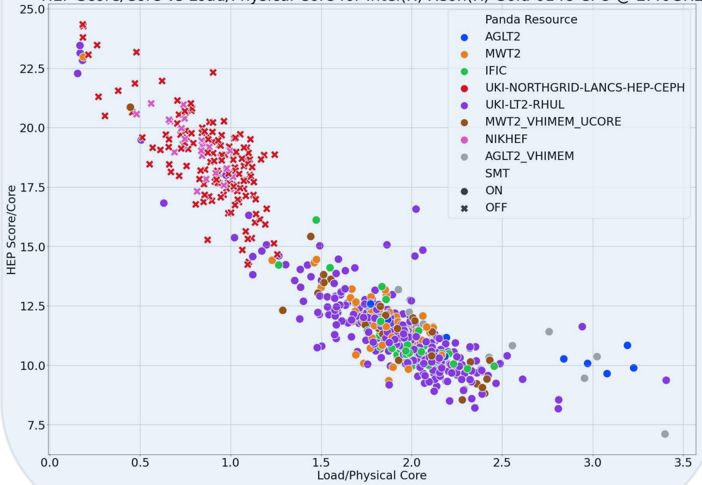
Correlation between performance of the machine and load per physical core

## ATLAS

HEP Score/Core vs Load/Physical Core for Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz

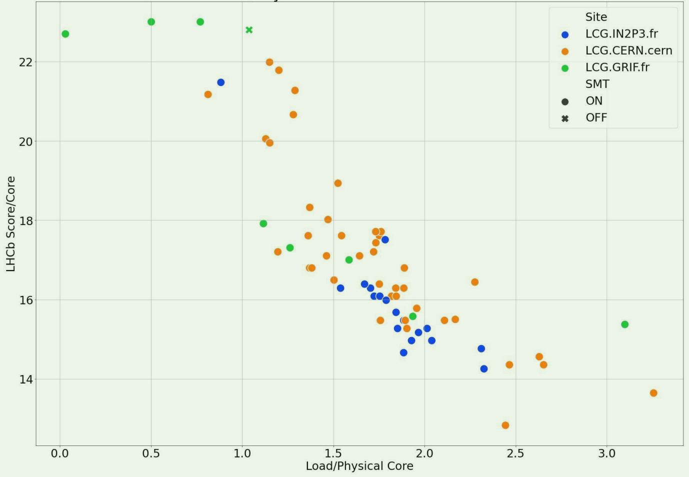


HEP Score/Core vs Load/Physical Core for Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz

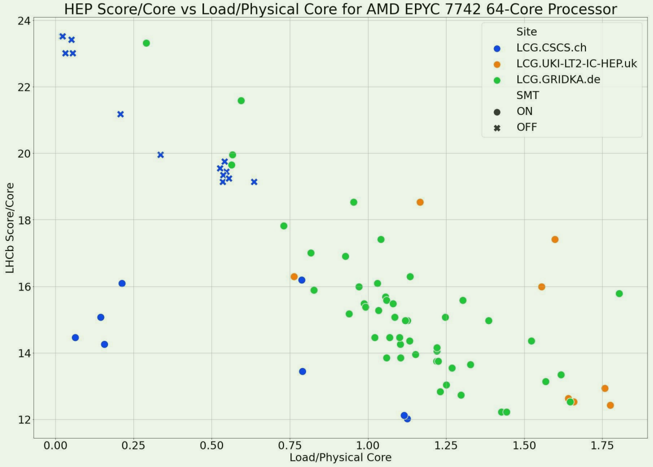


## LHCb

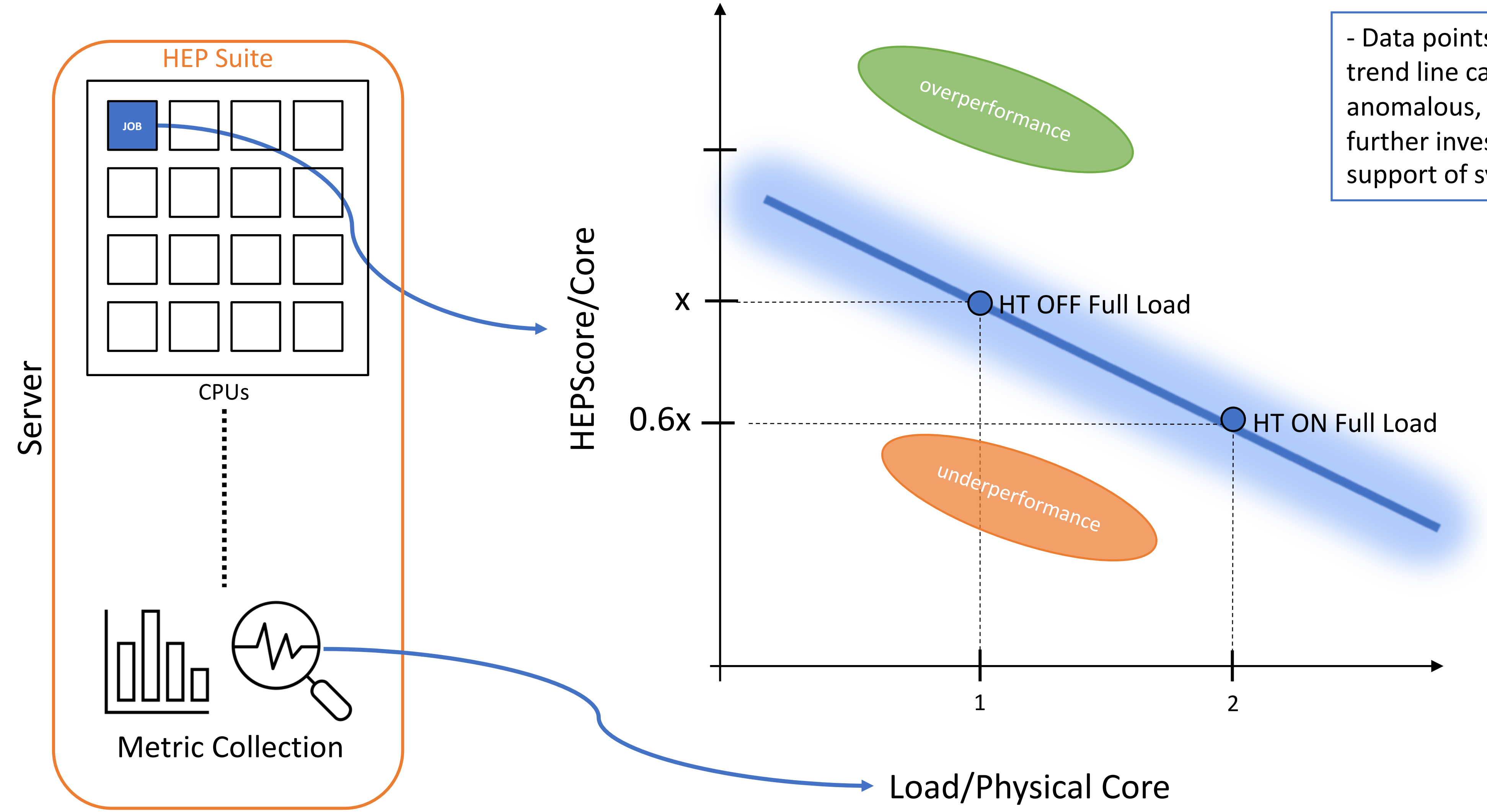
HEP Score/Core vs Load/Physical Core for AMD EPYC 7302 16-Core Processor



HEP Score/Core vs Load/Physical Core for AMD EPYC 7742 64-Core Processor

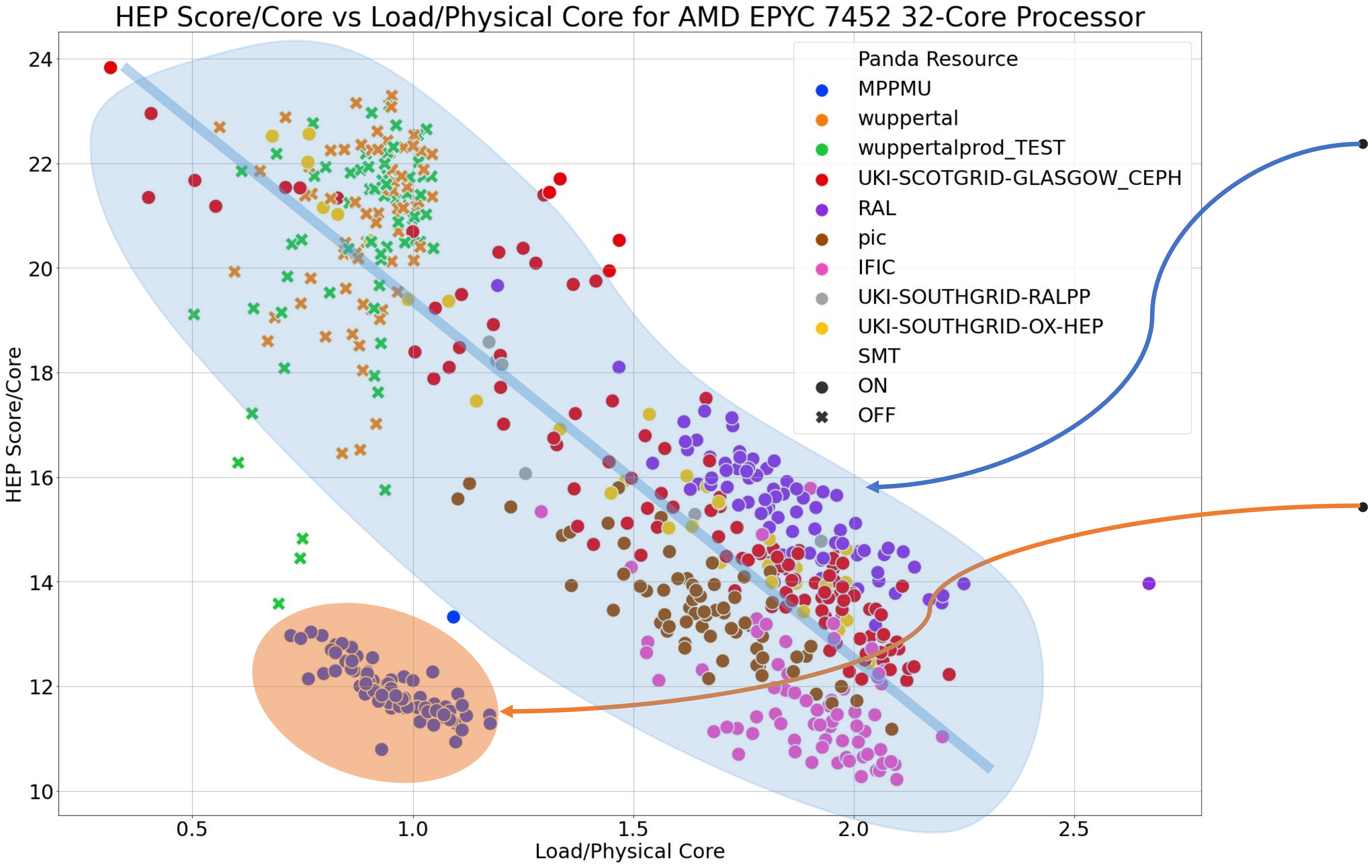


# Anomalous performance



- Data points outside the main trend line can be considered anomalous, and deserve further investigation with the support of sysadmins

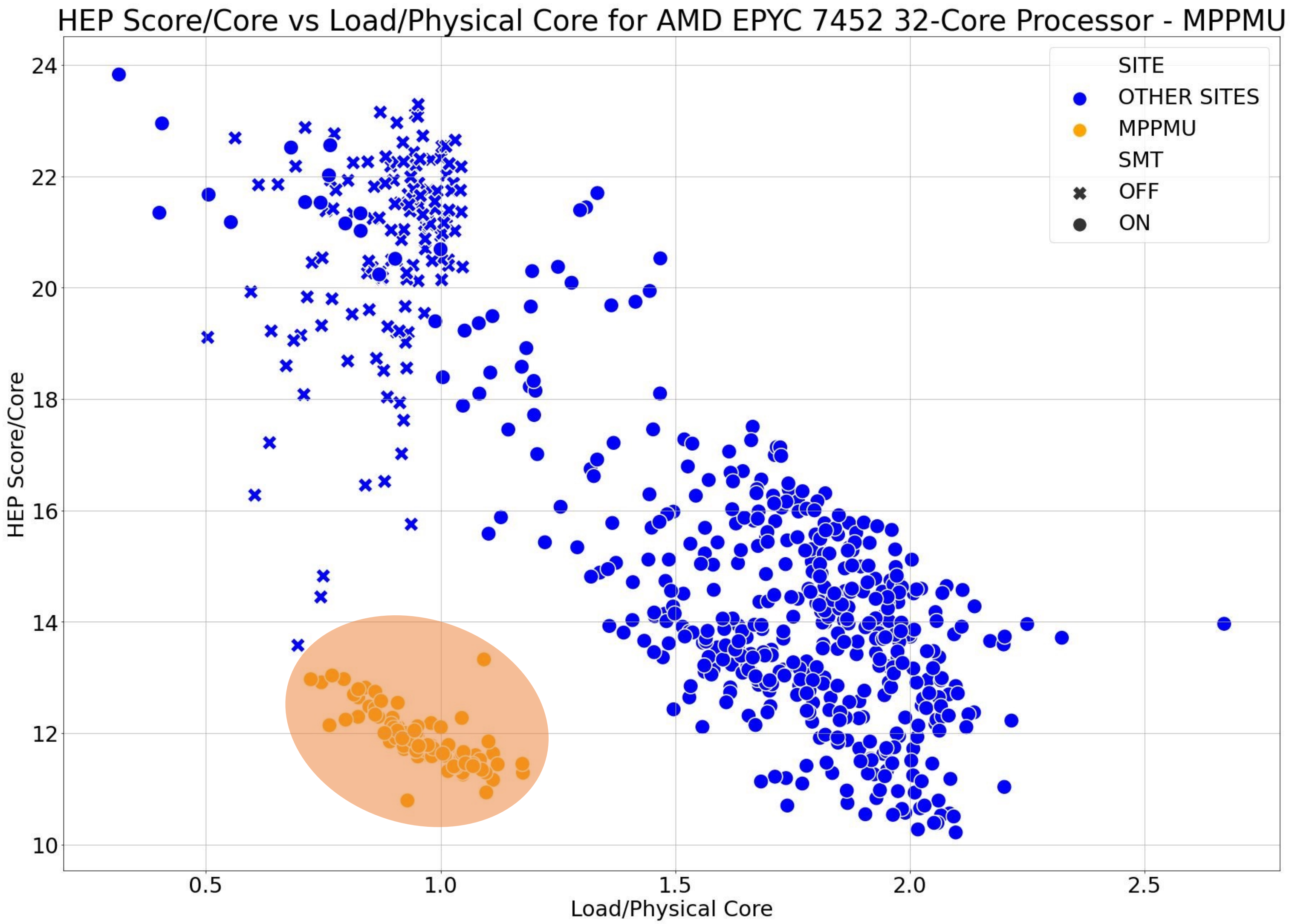
# Discriminate normal trend and anomalies



- Even if there is large spread in performance between sites, most of the data points follow the main trend

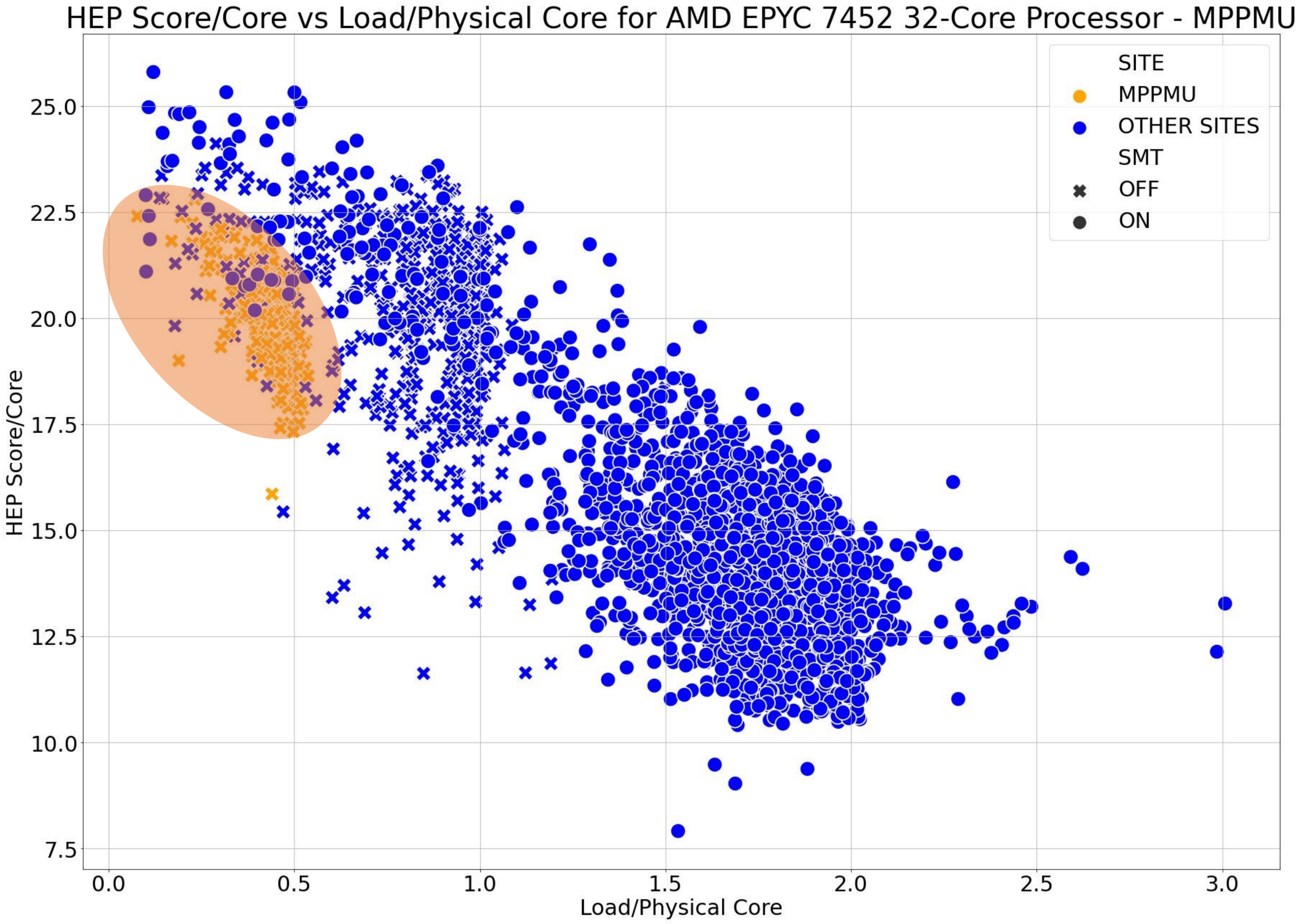
- Example of anomaly investigated with sysadmin (see next slide)

# Measurements before changes



- MPPMU performance in terms of score per core is low and comparable to other sites that have double load in the servers
- After site admin investigation, it was found out that servers had a wrong masking of the disabled cores while SMT was ON
    - Then, fix has been applied

# Measurements after changes

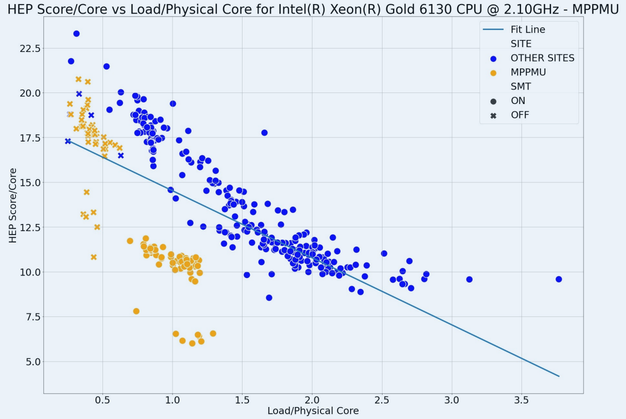


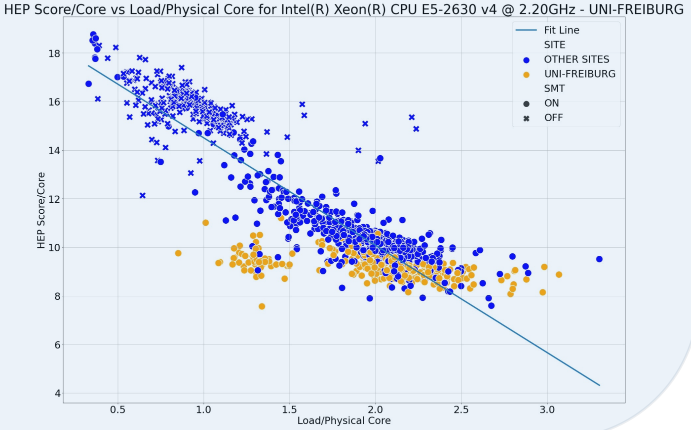
- Applied changes: SMT off fixed the problem with cores masking misconfiguration
- **MPPMU performance increased by 66%, load decreased**

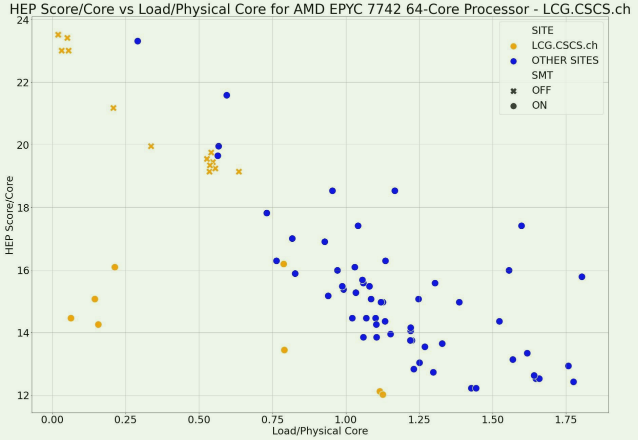
The analysis and issue resolution had a real impact on the site performance

# More examples

ATLAS


HEP Score/Core vs Load/Physical Core for Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz - MPPMU


HEP Score/Core vs Load/Physical Core for Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz - UNI-FREIBURG

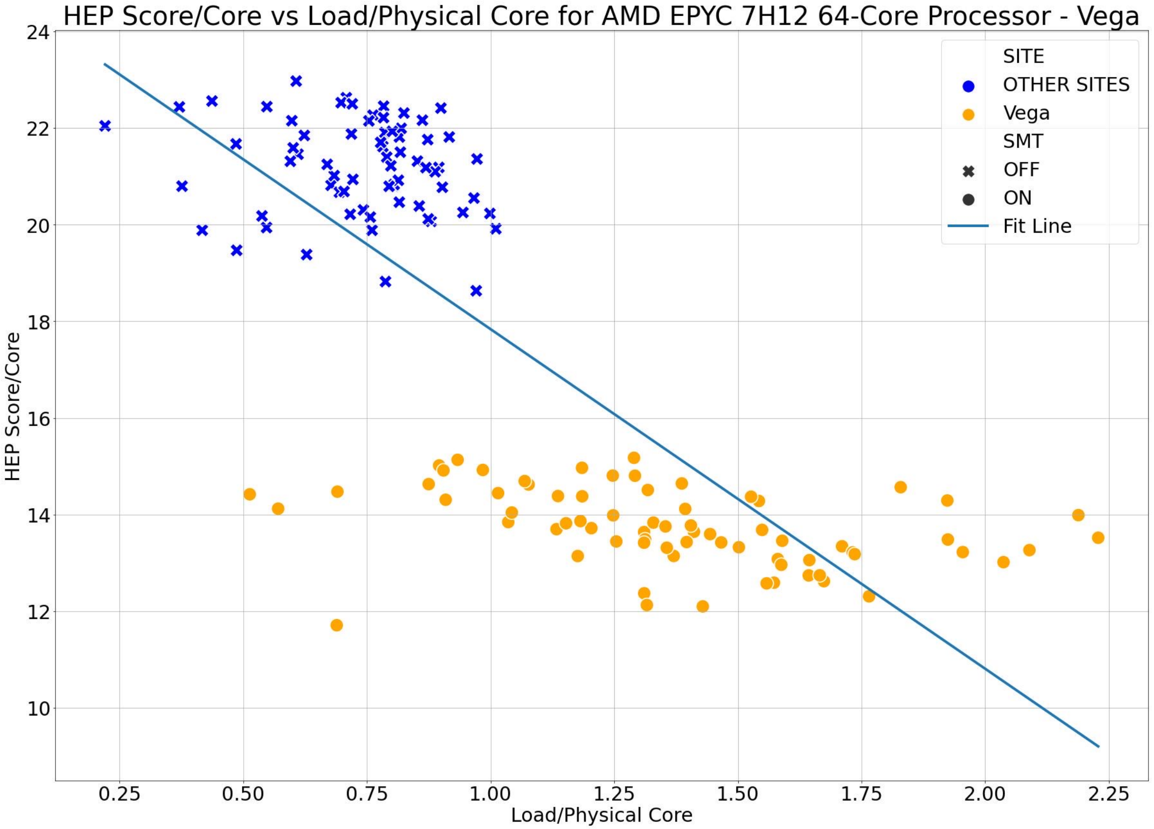

HEP Score/Core vs Load/Physical Core for AMD EPYC 7742 64-Core Processor - LCG.CSCS.ch

LHCb

We have contacted number of site admins to understand
how different configuration aspects impact on computing performance

# Deliberated choices

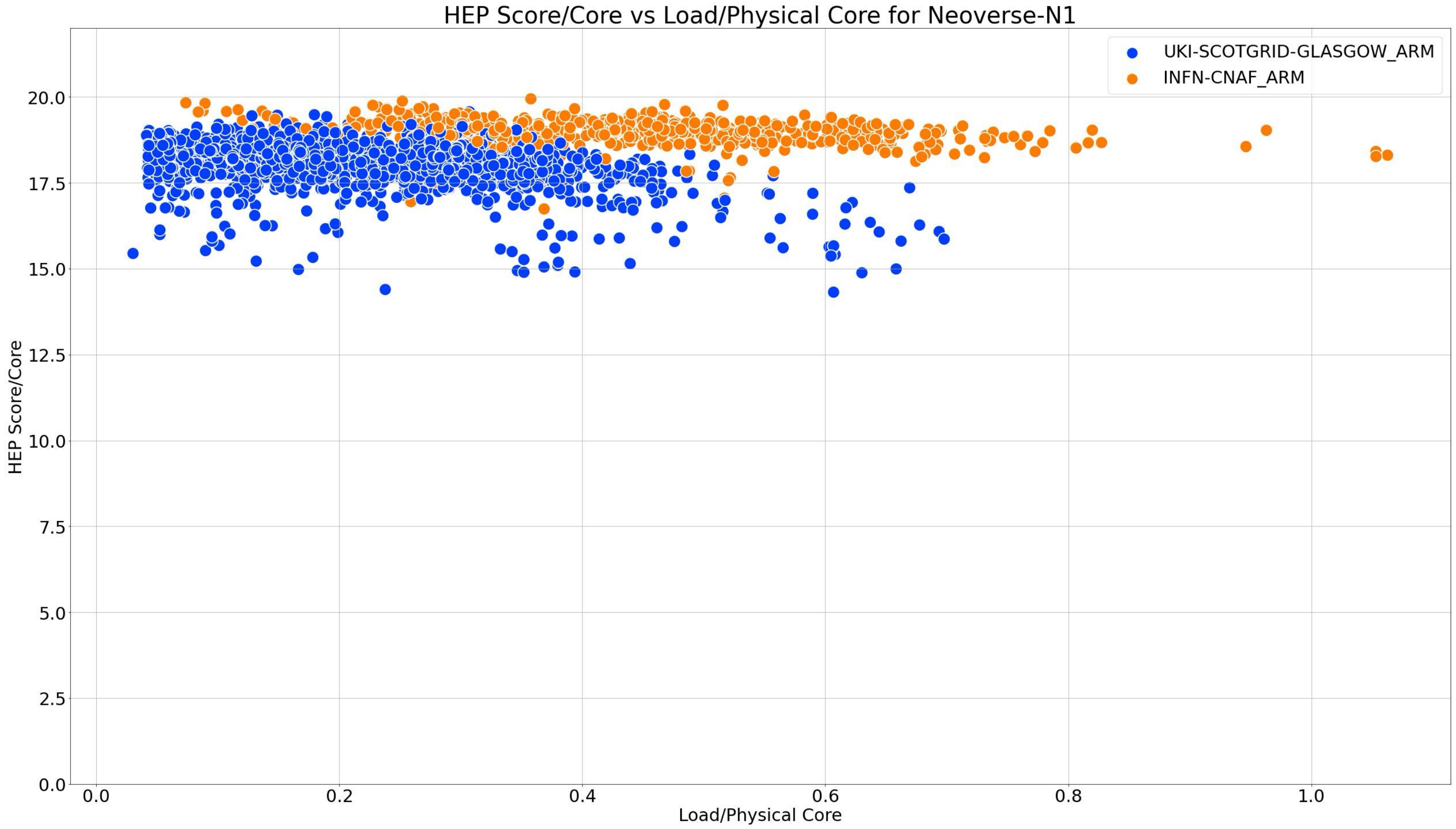
- Not all of the observed deviations from the model are issues, some of them are deliberated sites choices, for example:
  - VEGA is HPC queue: SLURM always allocate full physical cores (i.e. both sibling threads) to a job, to avoid interference with other job slots. As an effect the job performance is flat, not scaling with the server load.



# Last but not least ARM results

# ARM results

- 2 sites already exposing ARM servers for ATLAS:
    - CPU model: Ampere Altra at 160 cores
- The performance vs load does not apply here.
- ARM server does not include Hyper-Threading technology. The frequency stay stable during the run.
- Therefore we do not see correlation between load and performance as before.



# Work in progress

Metrics, including power consumption, memory usage, and other user-defined parameters, can provide us with valuable additional information. They offer insights into the potential impact of various configuration aspects on performance, aiding in the enhancement of related areas.

# Summary

## Infrastructure

- HEP Suite is a powerful, ready to use by different experiments thanks to its flexibility
- Enhanced HEP Suite was designed to measure metrics such as machine load, memory usage, power consumption and many more user-defined metrics

## Analysis

- Model of HEPScore vs Load allow us to fix misconfiguration issues and to understand how different configuration aspects affect the performance
- ARM does not have HT, therefore there is no correlation between load and performance, but other dependencies can be found

# Q&A