CPU BENCHMARKING IN THE ALICE EXPERIMENT

Students:
Avram Alexandru
Garaiman Andres

Supervisors:
Costin Grigoras
Latchezar Betev

PURPOSE

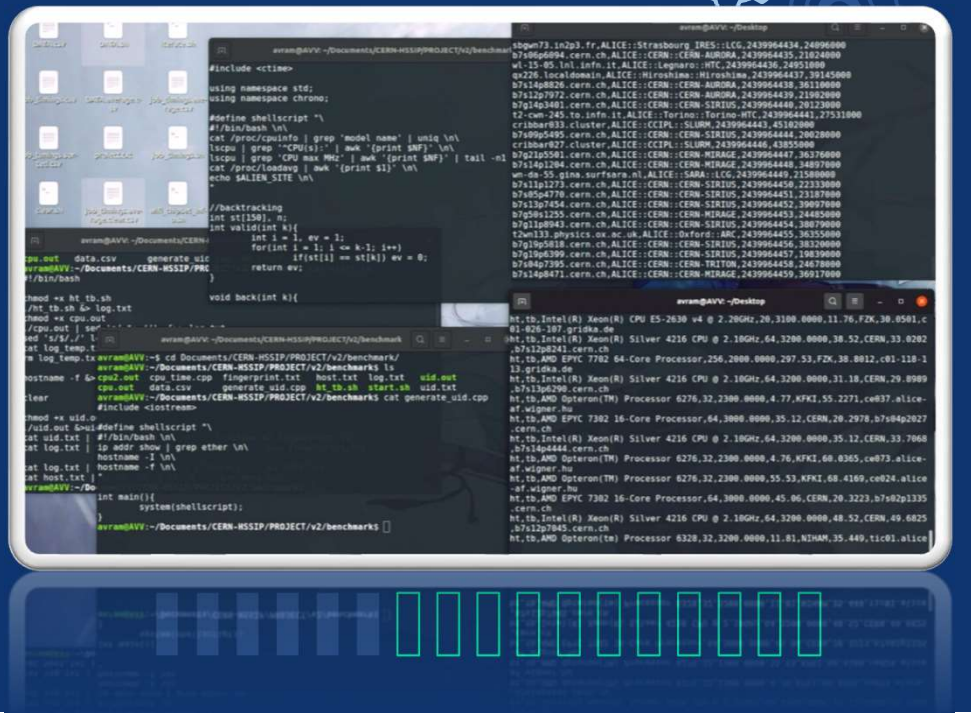
- Our project aims to study the infrastructure and the different types of CPUs in the ALICE GRID, by creating a benchmark that will then be compared with the MonteCarlo simulation jobs execution time;

PROJECT DESCRIPTION

- Collect statistics on the CPU types available to ALICE Grid jobs
- Implement a CPU benchmark that would evaluate the performance of those cores
- Run it on the distributed computing infrastructure
- Correlate the CPU model with the results of the benchmark

BENCHMARK

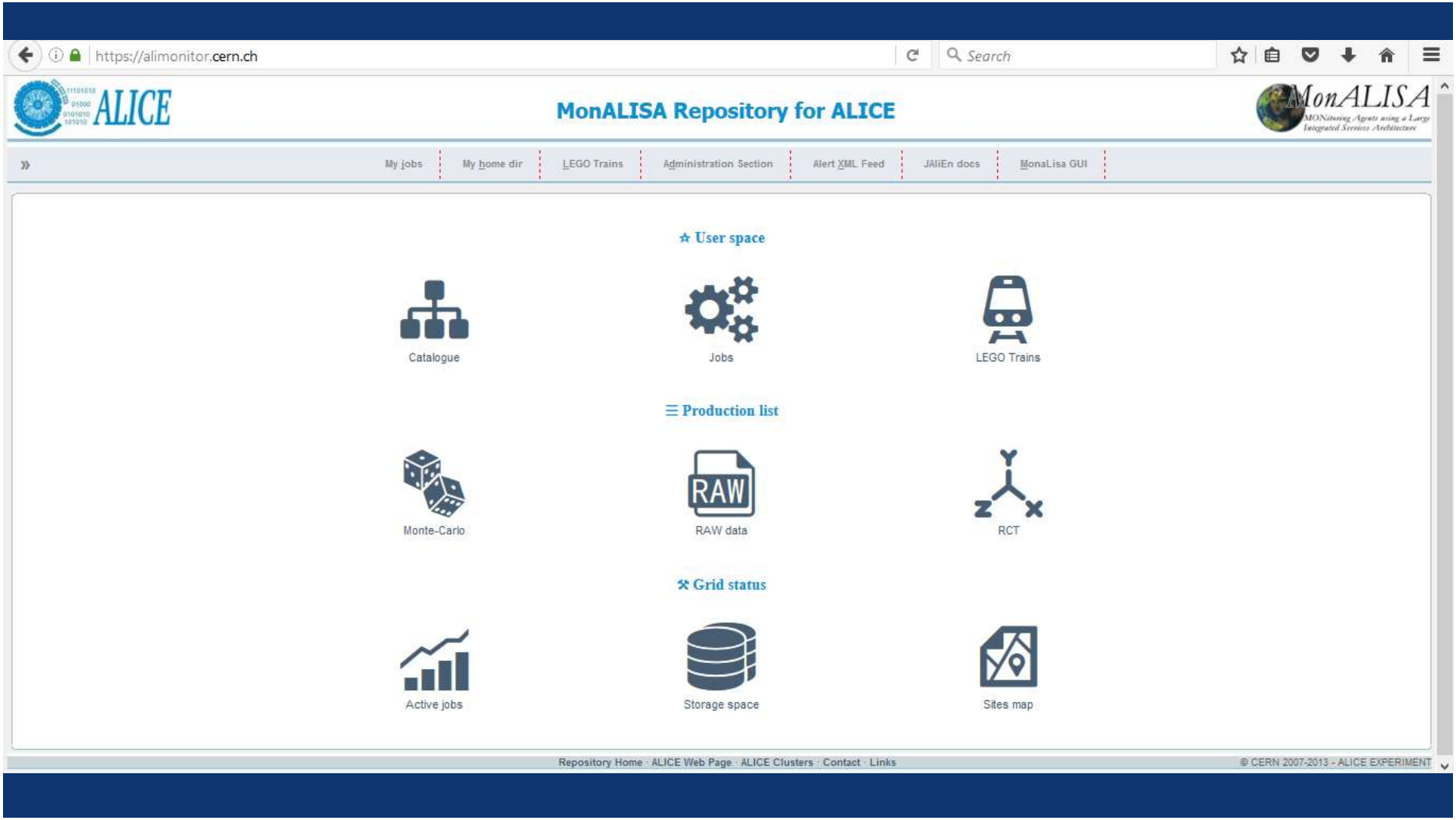
- In order to create the benchmark, we wrote a C++ program based on the backtracking algorithm and we analysed the time needed for computation, for each CPU model available.



DATA COLLECTION

- To collect the data needed for the CPU performance analysis, we ran around 400,000 jobs on the ALICE Grid infrastructure, and (after some bugs and errors) we ended up with 100,000 viable data points.
- The collected data consists of:
  - CPU model
  - No. of cores
  - Max. Frequency
  - CPU load
  - Execution Site
  - Computing time (result of the benchmark)

- Since our benchmark measures the duration of execution of jobs, the smaller the value -> the better the CPU is.


https://alimonitor.cern.ch
Search
ALICE
MonALISA Repository for ALICE
MonALISA
MONitoring Agents using a Large Integrated Services Architecture
My jobs
My home dir
LEGO Trains
Administration Section
Alert XML Feed
JAliEn docs
MonaLisa GUI
User space
Catalogue
Jobs
LEGO Trains
Production list
Monte-Carlo
RAW
RAW data
RCT
Grid status
Active jobs
Storage space
Sites map
Repository Home · ALICE Web Page · ALICE Clusters · Contact · Links
© CERN 2007-2013 - ALICE EXPERIMENT

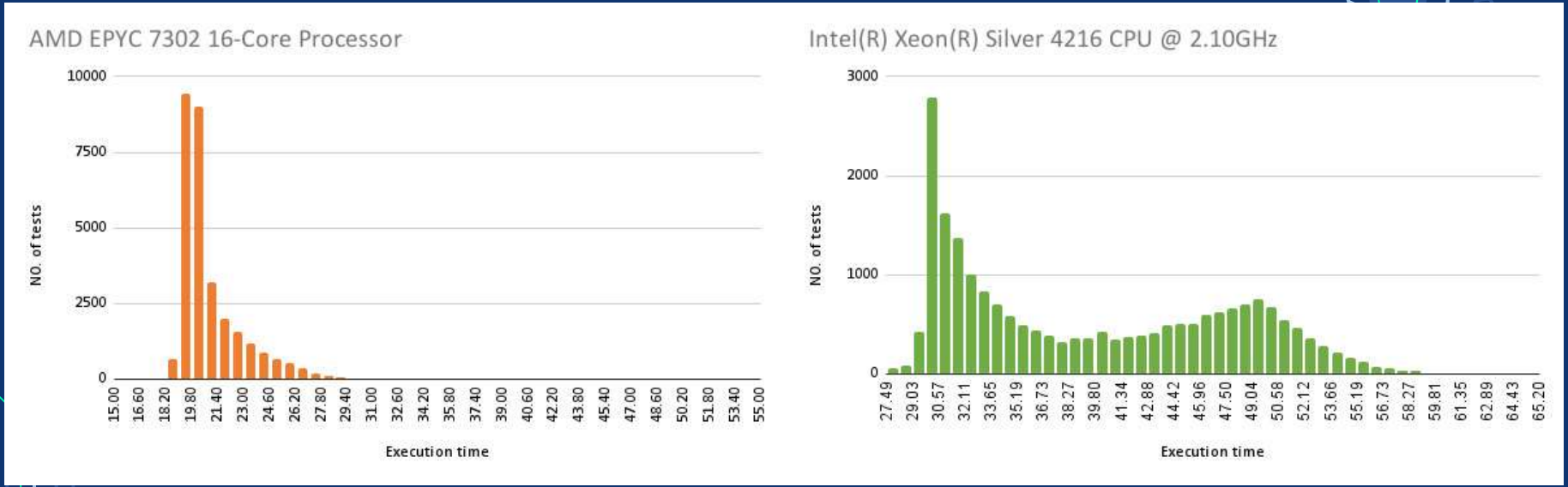
STATISTICS

After we finished collecting data, we started working on the following statistics:

- Comparison between the most used Intel CPU model and the most used AMD CPU model;
- Job distribution on different sites around the world;
- Correlation between our benchmark results and MonteCarlo simulations;

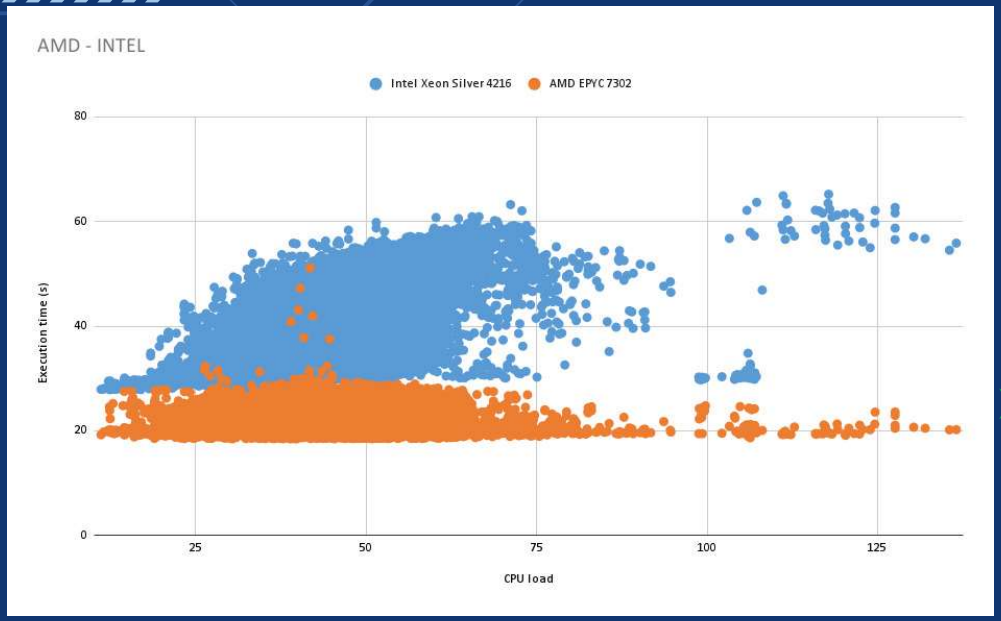
AMD VS. INTEL

Comparison between the most used CPU models



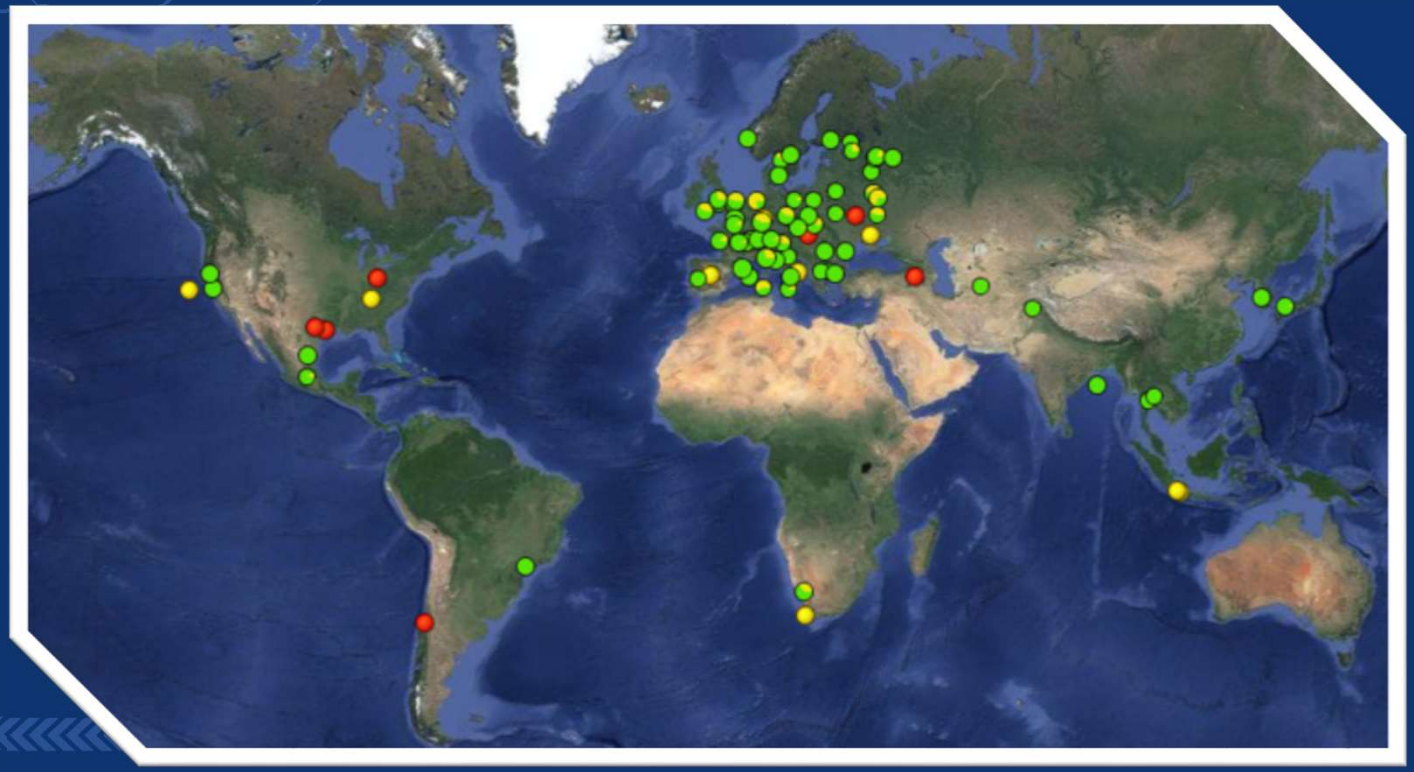
OUR INTERPRETATION

- Firstly, we noticed that our benchmark returned similar execution time on all tests performed by the AMD processors, while on the Intel CPUs there was a wide distribution of the duration values;
- We think this is due to the different architectures of the two CPU models: the AMD model has a stable behaviour, while the Intel model is strongly influenced by the Turbo Boost technology, because it changes the maximum frequency and how the data is processed according to the given job;

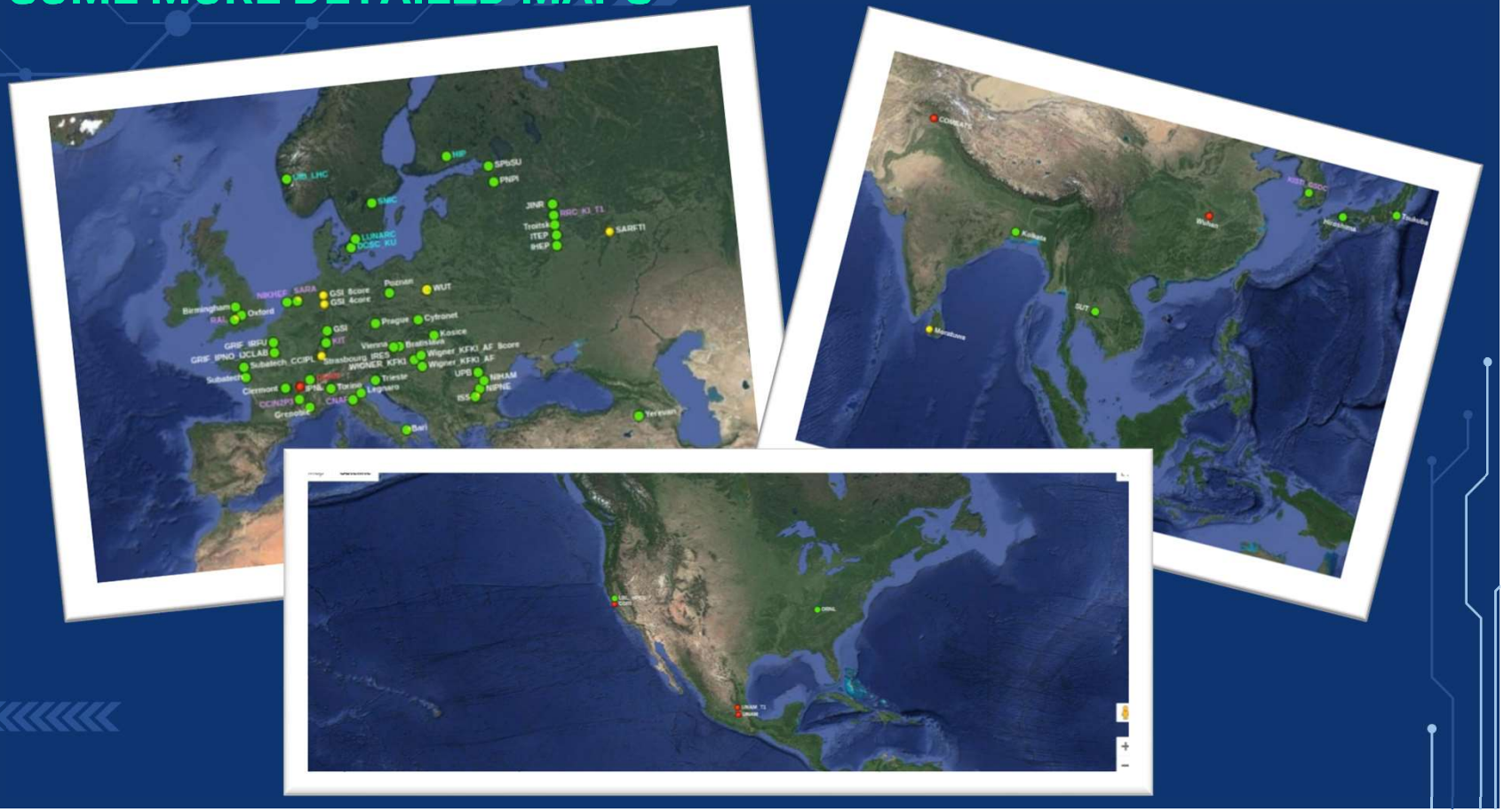


- This plot shows the result of our benchmark, correlated with the CPU load;
- The AMD processor clearly outperformed the Intel one, because it maintained a stable execution time with the increase of workload;
- On the other hand, for the Intel Xeon Silver 4216 CPU, the performance time varies a lot depending on the workload;

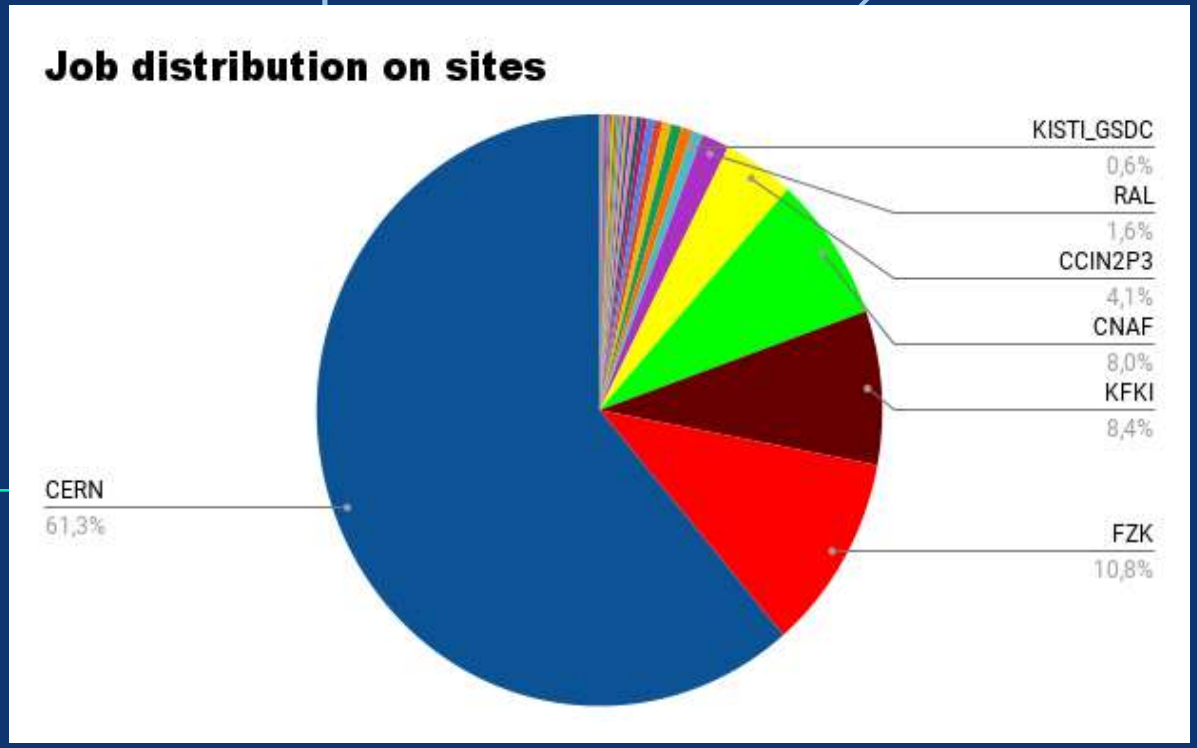
A VISUALISATION OF THE ALICE SITES WORLDWIDE



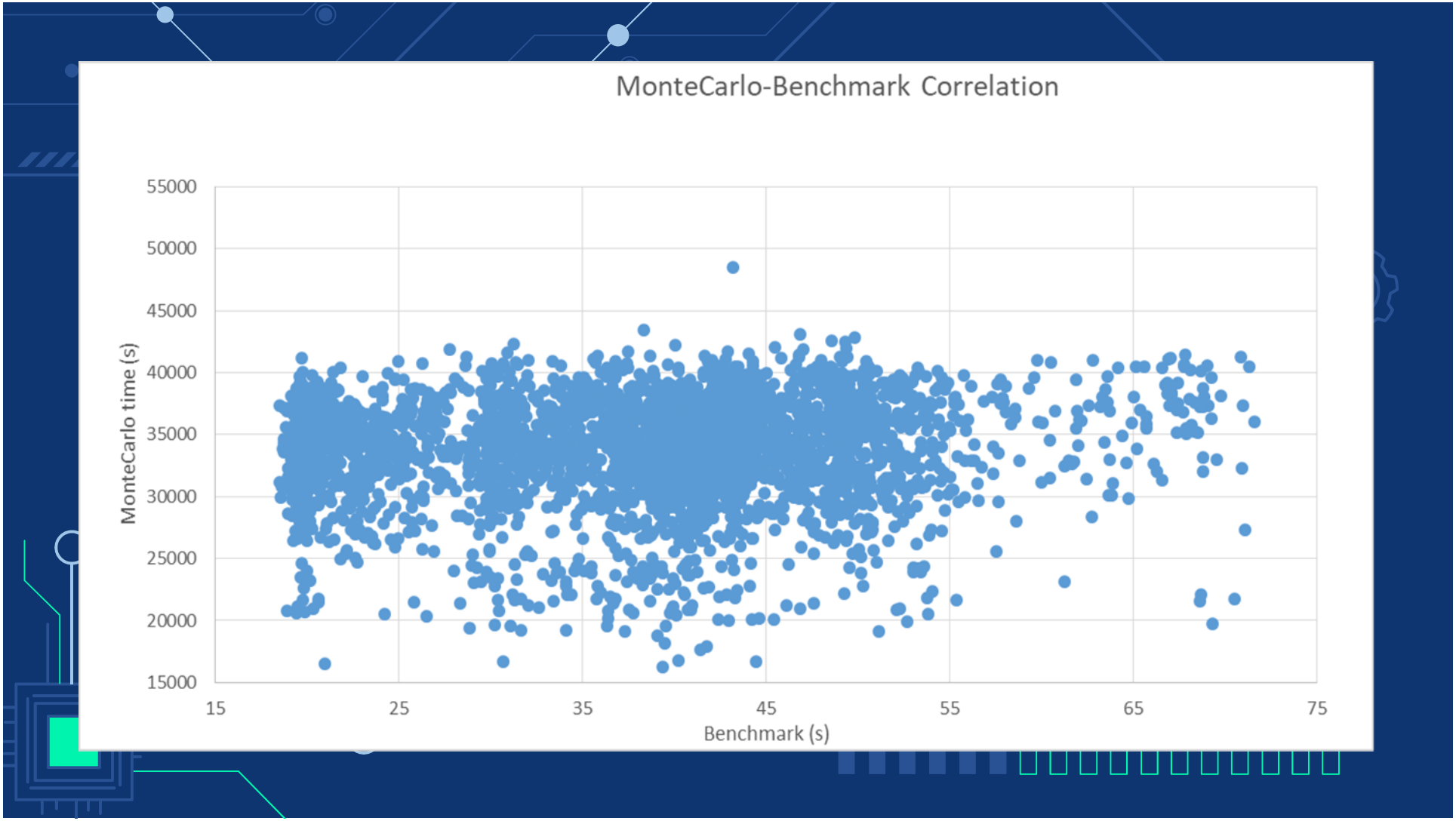
...AND SOME MORE DETAILED MAPS



ALICE GRID JOBS DISTRIBUTION



In total, we ran around 100,000 jobs in 36 different sites around the world, on 90 different CPU models.


MonteCarlo-Benchmark Correlation
MonteCarlo time (s)
55000
50000
45000
40000
35000
30000
25000
20000
15000
15
25
35
45
55
65
75
Benchmark (s)

CONCLUSIONS

- Why can’t we see an obvious correlation?
  - One reason is that our benchmark uses a rather simple code and tests the performance on one single core. It is best used for analysing the performance of “brute force”-type applications and other single-core processes. That is why our results may differ from others produced by more specialised benchmarks.
  - Another explanation is that MonteCarlo is a complex simulation that runs on multiple cores and involves more complicated algorithms, making use of the advanced features of the CPUs, like Turbo Boosting.

- Not only that, but MonteCarlo uses other components of the system, too. For example, it allocates and uses more memory (which in turn leads to more page transfers between the memory and the CPU);
- Another argument that we can give is that the code used for the simulation is very large and it doesn’t fit inside the CPU cache, which causes a constant loop of input and output;

WHAT HAVE WE LEARNED?

- ✓ We think that the most valuable lesson here is that research of any kind implies a lot of work, patience, persistence and not giving up. Even though you might not get your expected results, you should always continue your exploration, improving your methods all along the way.
- ✓ We also learned more about the ALICE Experiment and its computing infrastructure.
- ✓ Last, but not least, we were introduced to the world of CERN and we had a great time discovering its people and its technologies.

SPECIAL THANKS

- We want to thank our supervisors, Costin Grigoras and Latchezar Betev, for all their support.
- We also want to thank the amazing people that we met here and, let's not forget, the organizing team that made all of this possible.
- We hope to see you again soon!!!.....