JOINT INSTITUTE FOR NUCLEAR RESEARCH
Veksler and Baldin laboratory of High Energy Physics

FINAL REPORT ON THE INTEREST PROGRAMME

Data visualization service for computing jobs analytics

Supervisor:
Igor Pelevanyuk

Student:
Daniel Čampiš, Czech Republic
Palacký University Olomouc

Participation period:
05 October - 13 November, 2020, Wave 1

Dubna, 2020

Introduction

The goal of this project was to develop a graphical visualization from log information about processes computing scientific models. Those processes represent simulation models of nuclear particles decay inside of a controlled environment of a particle accelerator. Statistics about the computation jobs were logged into a database. Because of a demand to track the system and its performance a necessity to visualize job records arose. Goals of the development is to obtain records about each job from a remote database and process them into a local database. The local database is refreshed daily to keep up to date state. The actual visual part is represented by a browser application based on JavaScript programming language. The application consists of a chart which compares a time how long each job ran to a benchmark score of a processor's core where the job was processed. Other features enable to filter data by many selectors.

Project goals

- a remote database mirroring on daily basis
- set up a local database with structure suitable for the project
- visualize database entires in a chart
- randomize values on x axis +/- 0.05 to avoid aggregation
- additional chart properties: zoomability, custom size of chart's markers
- filter database entries by chosen selectors: site, time range, exact time, user, status
- user friendly UI

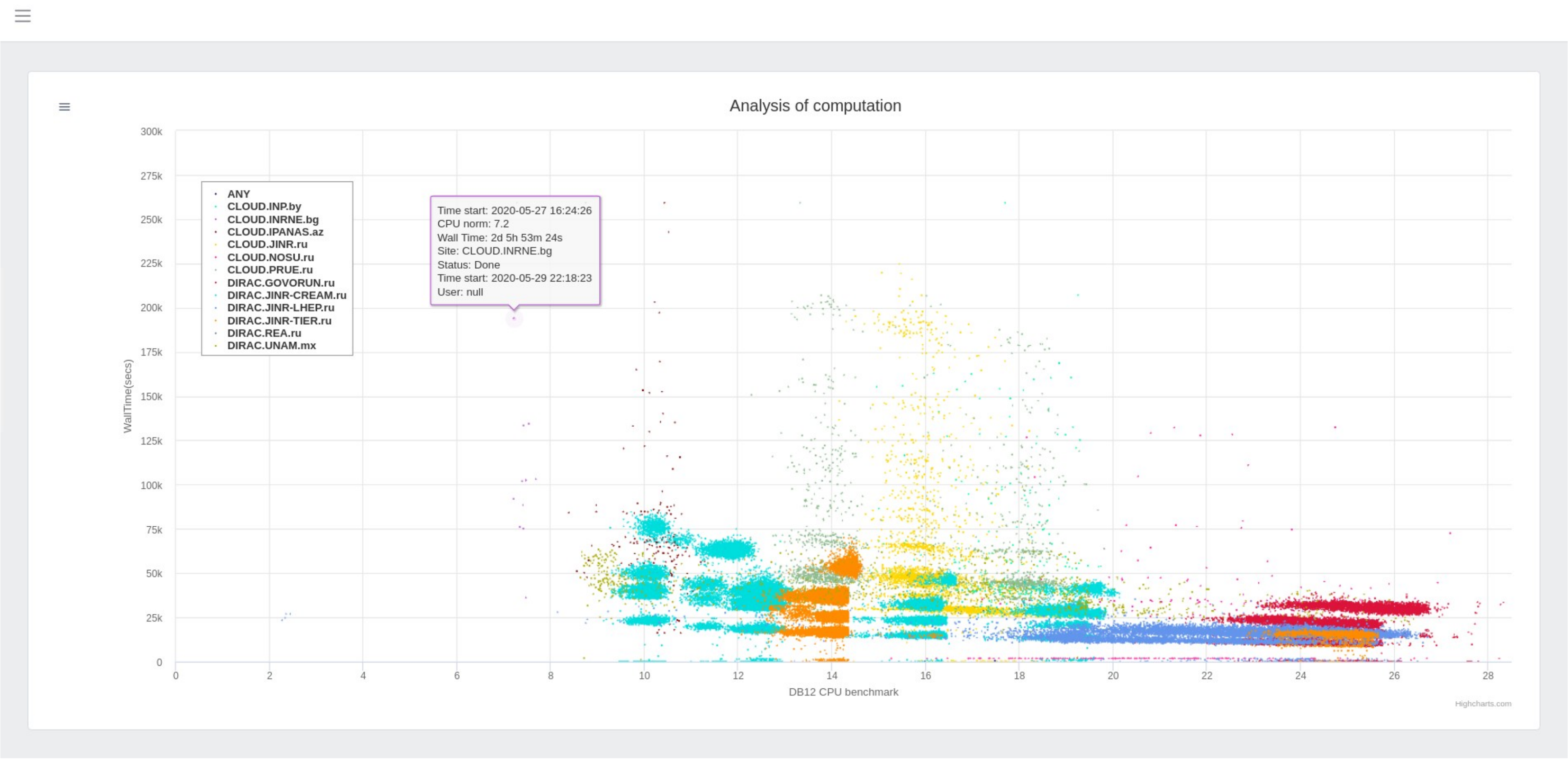
Scope of work

The project was led by the supervisor Igor Pelevanyuk. Tasks were consulted weekly over web calls and desktop sharing. Results were published on a private web server and a Gitlab repository.

Methods

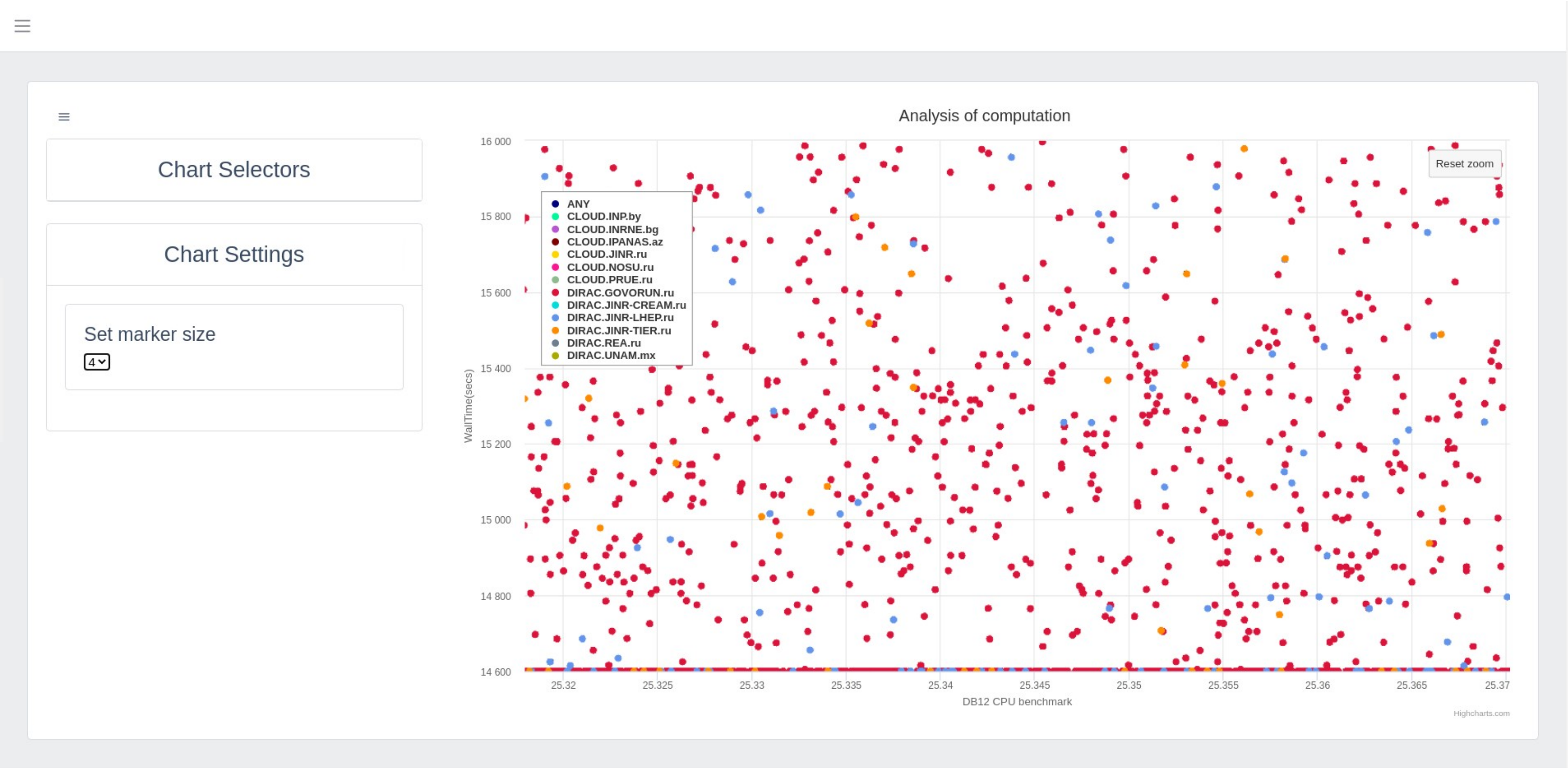
Methods used on the project were restricted to conventional web application development techniques.

Figures/diagrams/plots



Time start: 2020-05-27 16:24:26
CPU norm: 7.2
Wall Time: 2d 5h 53m 24s
Site: CLOUD.INRNE.bg
Status: Done
Time start: 2020-05-29 22:18:23
User: null

Screenshot 1 - A landing page of the web application with an example of a process information viewed in tooltip.



Set marker size
4

Screenshot 2 — A zoomed segment of the chart with enlarged marker size.

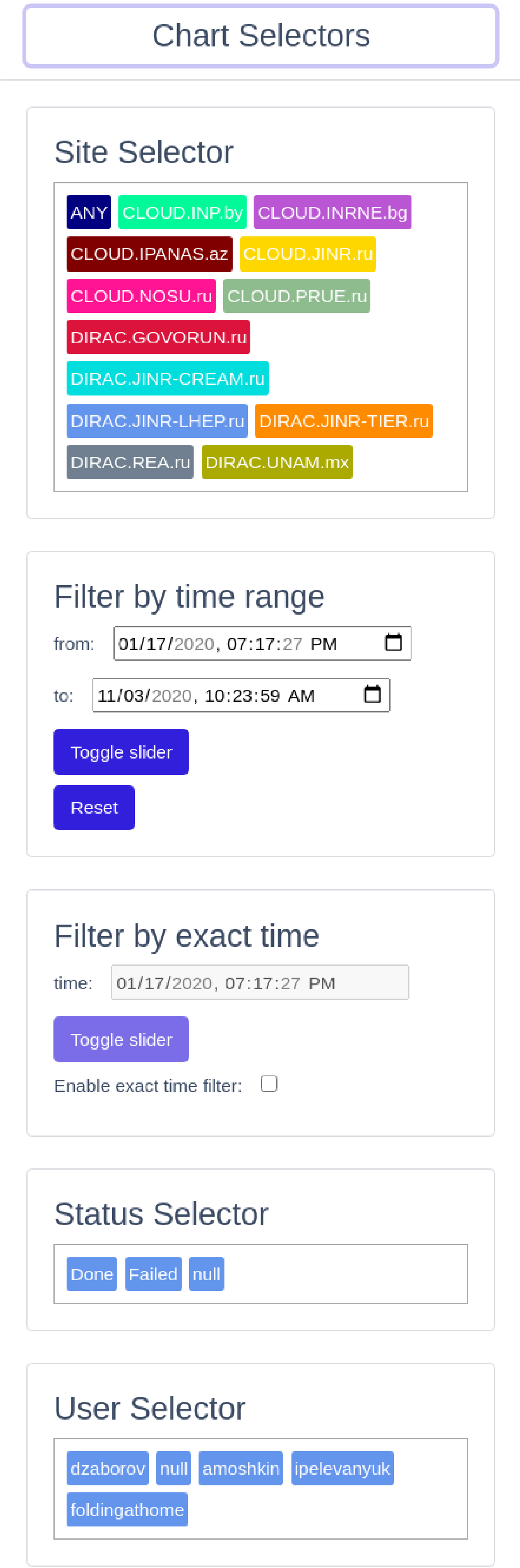


Chart Selectors

Site Selector

ANY CLOUD.INP.by CLOUD.INRNE.bg CLOUD.IPANAS.az CLOUD.JINR.ru CLOUD.NOSU.ru CLOUD.PRUE.ru DIRAC.GOVORUN.ru DIRAC.JINR-CREAM.ru DIRAC.JINR-LHEP.ru DIRAC.JINR-TIER.ru DIRAC.REA.ru DIRAC.UNAM.mx

Filter by time range

from: 01/17/2020, 07:17:27 PM

to: 11/03/2020, 10:23:59 AM

Toggle slider

Reset

Filter by exact time

time: 01/17/2020, 07:17:27 PM

Toggle slider

Enable exact time filter: ☐

Status Selector

Done Failed null

User Selector

dzaborov null amoshkin ipelevanyuk foldingathome

Screenshot 3 — A filetring column with available selectors.

Results

In the time of writing those papers, the application is hosted on the private web server powered by Raspberry Pi 3 board with specifications:

- processor: ARMv7, 32 bits. 700 - 1400 MHz
- memory: 1GB LPDDR2 (900 MHz)
- networking: 10/100 Ethernet
- storage: microSD card 64GB

Main bottlenecks of the solution is an undersized network bandwidth from a side of a internet provider and a slow reading speed of microSD card. Those are limiting factors to lower the web page load time which is approx. more then 10 sec.

The local database run on the Raspberry Pi server and is provided by InfluxDB. A custom script upgrades database entires from a publicly available csv file on every day. The script make sure there are no duplicates or corrupted entries imported and uploaded to local database.

An user interface is created as single page web application based on ReactJS framework and MobX state manager. Source code is bundled by Webpack and served by Node.js server. Once the application files are fetched an initial database request is processed and all local database records are copied into the application. The following view of filtered data is processed only in a browser without additional database requests.

The chart is represented by a two dimensional comparison of the wall time and the CPU benchmark score. Each database record is rendered as a scattered point.

Other features of the chart is data filtering by site, time range, exact time, user, status.

Conclusion (discussions, prospects)

Goals introduced at the begging were fulfilled but meantime the remote database structure changed. At the present database can't be updated and a future work should be focused on restructuring the remote database.