# ENS 491 – Graduation Project (Design)

# Progress Report I

**Project Title: Real-Time Super Computer Monitoring**

**Group Members:**

**Alperen Yıldız**

**Bilgehan Çağıltay**

**Elif Cemre Durgut**

**Supervisor: Kamer Kaya**

**Date: 07/01/2023**

## 1. PROJECT SUMMARY

The real-time supercomputer monitoring project (SuperTwin) is a part of SparCity, which is an international project where 6 partners from 4 different countries collaborate with 2.6 M € funding by the European High-Performance Computing Joint Undertaking (EuroHPC JU).

SuperTwin aims to create an automated monitoring tool by utilizing machine learning. The goal of this tool is to monitor for, recognize, and respond to reductions in performance. Within this project, the objective is not only to create a tool that can be utilized to view the performance of supercomputers with respect to their configurations but also to extract data and shed further light on performance anomalies associated with some configurations and cache accesses.

SuperTwin gathers data from supercomputers and creates a structured dashboard representation by managing data input from a collection of tools. Some capabilities of SuperTwin are knowledge retrieval, modeling, monitoring, and automated profiling. In the end, it is expected to optimize data retrieval and storage, improve the data monitoring visualizations, use machine learning to predict future performance, and detect anomalies.

The main drawbacks of existing supercomputer monitoring systems are

- Cost: Supercomputers are expensive to purchase and maintain. The cost of implementing and maintaining a monitoring system can add significantly to the overall cost.
- Complexity: Supercomputers are complex systems with many different components that need to be monitored, which can make the monitoring process itself complex.
- Scalability: As supercomputers grow in size and complexity, it can be difficult to scale the monitoring system to keep up with the increased demands.
- Integration with other systems: Supercomputers often need to integrate with other systems and technologies. It can be challenging to ensure that the monitoring system works seamlessly with these other systems.
- Accuracy: It is important for monitoring systems to be accurate in order to effectively identify and diagnose problems. However, achieving high levels of accuracy can be difficult, especially as supercomputers become more complex.

To address these issues, SuperTwin aims to extract data from standalone Linux servers and HPC clusters and process this large volume of data so that it can be used to train machine learning models. Additionally, the data is also processed and visualized to systems owners so as to clearly show what is currently happening within the system. The trained machine learning model will then be used to detect anomalies, provide insights on how to achieve better performance and energy efficiency, and report performance characteristics.

Furthermore, the tool will feature detailed documentation on its usability and features. This will be complemented by visual animations for providing a user-friendly insight into the inner workings of the system.

Our objectives for this project are as follows:

1. Extraction of data from a standalone Linux server and an HPC cluster
2. Visualization of the extracted data
3. Usage of the data to train machine learning models to detect anomalies, performance characteristics. Provide insights for better performance and energy efficiency
4. Documentation of the tool and visual animation to provide insight into its inner workings of it.

## 2. SCIENTIFIC/TECHNICAL DEVELOPMENTS

Since the submission of our previous Proposal Report, we have installed the necessary technologies, libraries, and dependencies, including MongoDB, Grafana, and InfluxDB, on our personal computers to enable us to execute the Supertwin project.

As previously mentioned in our last project report, one of our objectives was to create comprehensive documentation for the SuperTwin project, including instructions for installation on various operating systems and guidance on running and testing the project on supercomputers. To this end, we have created a readthedocs document in which we provide a detailed explanation of the project. You can reach the last version of the documentation via this link.

In order to accommodate users who utilize various types of operating systems, we have provided installation tutorials for systems such as Ubuntu, Manjaro, and MAC. We have also included potential warnings and notes in the documentation to address some errors that users may encounter.

In addition to the installation tutorial above, the documentation includes sections such as “How to contribute to the project?” in which we explained how to add to the repository, how to report found bugs or fixes, and also the “FAQ” section to answer the most commonly asked questions.

Additionally, the setup was automated using a series of bash and python scripts so that the setup process can proceed smoothly in any *nix operating system. The setup process works by calling a bash script which automatically detects on which operating system it is being run and adapts the setup process depending on the operating system while also maintaining a detailed log file.

For the test process, we conducted tests on new virtual machines from scratch to ensure that the installation commands and setup scripts work correctly. In the future, we plan to expand our tests by trying the setup scripts on new ENS491 students’ computers. This will serve us as a useful testing platform and also ease the installation process for new students.

For machine learning and anomaly detection parts, we will identify the critical variables among different parameters and run tests on data collected from supercomputers. We may create artificial anomalies to assess the ability of our system to detect them.

Also, the tested processors so far had Intel processors. To have more variety, we will test the project on machines with ARM processors as well. You can find this newly added task on our new project timetable in Section 5.

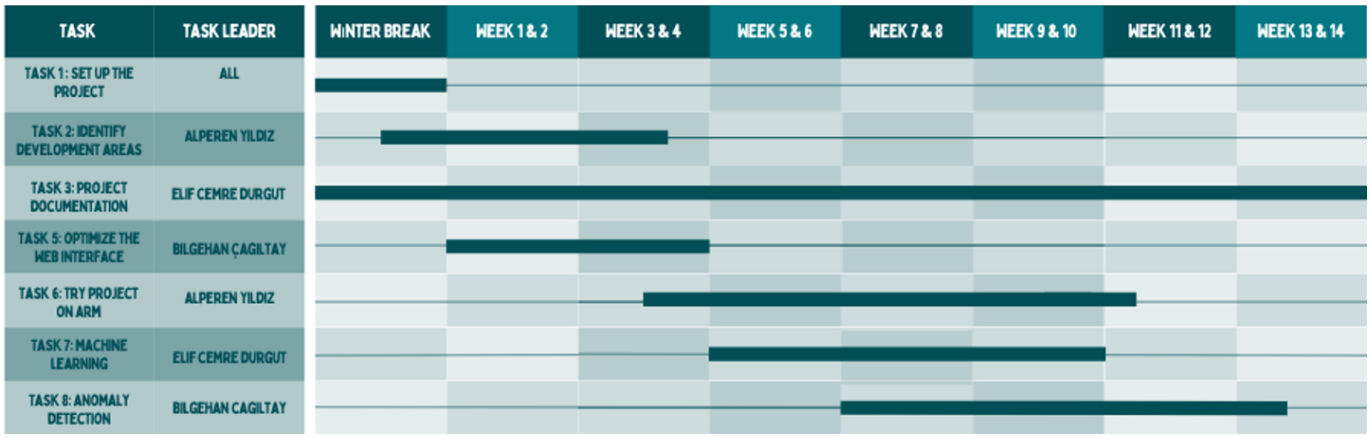
## 3. ENCOUNTERED PROBLEMS

As previously mentioned in our Proposal Report, we had eight main tasks planned for the project, including setting up the project, exploring the code, identifying development areas, creating a project tutorial, improving the web interface, and implementing machine learning and anomaly detection.

Since our last report, we have successfully completed the first four tasks. However, we encountered some challenges, particularly during the installation process, due to the complexity of the SuperTwin project and some version conflicts. Also, we encountered difficulties in finding reliable installation instructions for the required technologies online, which caused delays. To prevent future users from encountering the same issues we did, we have included the correct installation commands in our documentation.

According to our Gantt chart in the previous report, we were meant to have progressed to improving the web interface and testing machine learning algorithms on supercomputer data at this point, but we have fallen slightly behind schedule. To compensate for this, we shifted some tasks to our winter break to achieve our original goals on time.

## 4. TASKS TO BE COMPLETED UNTIL PROGRESS REPORT II



## 5. **REFERENCES**

SparCity (2021-2022). Deliverables. Taken from https://sparcity.eu/index.php/deliverables/deliverables/ on 08/01/2023.