

Aihemaiti G. – TEAM6

| | |
|-------------------------------|---|
| UI Design: | <ul style="list-style-type: none"> ➤ Designed the layout of the website ➤ Created a moodboard for colors and style ➤ Made the interface simple and easy to use |
| Front-End Development: | <ul style="list-style-type: none"> ➤ Built the front end using HTML CSS and JavaScript ➤ Added custom themes ➤ Worked on navigation and page layout |
| Testing and Debugging: | <ul style="list-style-type: none"> ➤ Tested pages to make sure everything looks correct ➤ Fixed styling bugs and layout issues ➤ Solved problems with CSS not updating |
| Deployment: | <ul style="list-style-type: none"> ➤ Deployed the website to the CS server ➤ Used Git and GitHub to manage code and updates ➤ Made sure all pages and links work |
| Impact: | <ul style="list-style-type: none"> ➤ Improved the look and usability of the website ➤ Helped create a clean final product |

Chimera Performance Metrics

Chimera Performance Metrics

Project Overview
This project focuses on collecting and displaying Chimera performance data in a clear and user-friendly dashboard.

Metrics
This section will show important system and GPU metrics such as CPU usage, memory usage, GPU utilization, temperature, and power consumption.

Documentation
Our task is to collect performance metrics from servers on the UMass Boston CS cluster and display them on a dashboard website. The project uses two main tools:

- Prometheus - software that reads data metrics from a server and stores them in a database.
- Grafana - software that displays the data stored by Prometheus as visual charts and graphs.

 In order for Prometheus to collect data, a node exporter must be deployed on each server. A node exporter is the program that takes live readings from the system - CPU usage, memory, system load, and more. It can only report what is happening right now and has no way of storing data on its own. This is where Prometheus comes in. Prometheus periodically reads from the node exporter and stores the results in its database, building a history of the system's performance over time.

So far, we have accomplished 4 main tasks:

- Set up virtual machines on CS server - The VM runs Ubuntu on the CS server. All group members have access to SSHing into the machine.
- Run node exporter on Balabog and Chimera - Balabog is the CS department's GPU server. Chimera is the cluster we are monitoring. We ran node_exporter on both, successfully loading their metrics.
- Prometheus data collection - Prometheus is successfully scraping data from the node exporters and storing it.
- Grafana display - Grafana is connected to Prometheus and correctly displaying the collected

Chimera Performance Metrics

Project Overview
This project focuses on collecting and displaying Chimera performance data in a clear and user-friendly dashboard.

Metrics
This section will show important system and GPU metrics such as CPU usage, memory usage, GPU utilization, temperature, and power consumption.

Documentation
Our task is to collect performance metrics from servers on the UMass Boston CS cluster and display them on a dashboard website. The project uses two main tools:

- Prometheus - software that reads data metrics from a server and stores them in a database.
- Grafana - software that displays the data stored by Prometheus as visual charts and graphs.

 In order for Prometheus to collect data, a node exporter must be deployed on each server. A node exporter is the program that takes live readings from the system - CPU usage, memory, system load, and more. It can only report what is happening right now and has no way of storing data on its own. This is where Prometheus comes in. Prometheus periodically reads from the node exporter and stores the results in its database, building a history of the system's performance over time.

So far, we have accomplished 4 main tasks:

- Set up virtual machines on CS server - The VM runs Ubuntu on the CS server. All group members have access to SSHing into the machine.
- Run node exporter on Balabog and Chimera - Balabog is the CS department's GPU server. Chimera is the cluster we are monitoring. We ran node_exporter on both, successfully loading their metrics.
- Prometheus data collection - Prometheus is successfully scraping data from the node exporters and storing it.
- Grafana display - Grafana is connected to Prometheus and correctly displaying the collected

Chimera Performance Metrics

Project Overview
This project focuses on collecting and displaying Chimera performance data in a clear and user-friendly dashboard.

Metrics
This section will show important system and GPU metrics such as CPU usage, memory usage, GPU utilization, temperature, and power consumption.

Documentation
Our task is to collect performance metrics from servers on the UMass Boston CS cluster and display them on a dashboard website. The project uses two main tools:

- Prometheus - software that reads data metrics from a server and stores them in a database.
- Grafana - software that displays the data stored by Prometheus as visual charts and graphs.

 In order for Prometheus to collect data, a node exporter must be deployed on each server. A node exporter is the program that takes live readings from the system - CPU usage, memory, system load, and more. It can only report what is happening right now and has no way of storing data on its own. This is where Prometheus comes in. Prometheus periodically reads from the node exporter and stores the results in its database, building a history of the system's performance over time.

So far, we have accomplished 4 main tasks:

- Set up virtual machines on CS server - The VM runs Ubuntu on the CS server. All group members have access to SSHing into the machine.
- Run node exporter on Balabog and Chimera - Balabog is the CS department's GPU server. Chimera is the cluster we are monitoring. We ran node_exporter on both, successfully loading their metrics.
- Prometheus data collection - Prometheus is successfully scraping data from the node exporters and storing it.
- Grafana display - Grafana is connected to Prometheus and correctly displaying the collected

Chimera Performance Metrics

Project Overview
This project focuses on collecting and displaying Chimera performance data in a clear and user-friendly dashboard.

Metrics
This section will show important system and GPU metrics such as CPU usage, memory usage, GPU utilization, temperature, and power consumption.

Documentation
Our task is to collect performance metrics from servers on the UMass Boston CS cluster and display them on a dashboard website. The project uses two main tools:

- Prometheus - software that reads data metrics from a server and stores them in a database.
- Grafana - software that displays the data stored by Prometheus as visual charts and graphs.

 In order for Prometheus to collect data, a node exporter must be deployed on each server. A node exporter is the program that takes live readings from the system - CPU usage, memory, system load, and more. It can only report what is happening right now and has no way of storing data on its own. This is where Prometheus comes in. Prometheus periodically reads from the node exporter and stores the results in its database, building a history of the system's performance over time.

So far, we have accomplished 4 main tasks:

- Set up virtual machines on CS server - The VM runs Ubuntu on the CS server. All group members have access to SSHing into the machine.
- Run node exporter on Balabog and Chimera - Balabog is the CS department's GPU server. Chimera is the cluster we are monitoring. We ran node_exporter on both, successfully loading their metrics.
- Prometheus data collection - Prometheus is successfully scraping data from the node exporters and storing it.
- Grafana display - Grafana is connected to Prometheus and correctly displaying the collected

| | |
|-------------------|---|
| GitHub: | https://github.com/B6ChiP/chimera-dashboard |
| Artifacts: | /cs410/s26/hdeblois/GROUP2/longproj02/t6/ |
| Live Site: | https://www.cs.umb.edu/~hdeblois/cs410/longproj02/t6/ |

T-Shaped skills

GitHub ★ Deployment (CS Server)
★ Testing ★ Debugging ★ Team Communication

| |
|-------------------|
| UI / Frontend |
| HTML CSS JS |
| Theme System |
| Responsive Design |
| UI Debugging |

Growth

| |
|-------------------------|
| Grafana Basics |
| familiar Chimera System |
| Frontend |
| Backend Concepts |
| Faster Debugging |
| Cleaner Code |
| Deployment Confidence |

Additional Comments or Plans.

Plans:

Improve backend skills

Learn more about system metrics and Grafana

Continue developing the Chimera project

Comments:

Improved frontend and UI skills

Learned a lot from teammates with different strengths

Improved collaboration and communication skills

Research paper: </courses/cs410/s26/gulsum/cs410/rpaper.txt>

