Project 3 Report

Xinyu Tian, Zesen Zhuang

March 5, 2023

0.1 Project Structure

```
└─ dashboard ....................... dashboard source code for task 2
   ├─ Dockerfile
   ├─ app.py
   ├─ data.json
   ├─ requirements.txt
   ├─ utils
   │  └─ helper_functions.py
   └─ wsgi.py
└─ k8s ..................... kubernetes configurations for task 1, 2, 3
   ├─ base ................ using provided serverless runtime, task 1, 2
   │  ├─ kustomization.yaml
   │  ├─ outputkey-config.yaml
   │  ├─ pyfile-config.yaml
   │  └─ serverless-deployment-course.yaml
   ├─ overlays
   │  └─ dev ........... Using implemented serverless runtime, task 3
   │     ├─ deployment.yaml
   │     ├─ kustomization.yaml
   │     ├─ pyfile
   │     └─ service.yaml
   └─ pyfile-config.yaml
└─ makefile
└─ ourmodule.py ............................. usermodule for task 1
└─ requirements.txt
└─ serverless-runtime ..... serverless runtime source code for task 3
   ├─ Dockerfile
   ├─ main.py
   ├─ requirements.txt
   ├─ tests
   │  └─ test_redis_connection.py
   ├─ usermodule_dumour.py
   └─ utils
      ├─ context.py
      └─ redis_connection.py
```

The project structure is shown in the above tree which illustrates the code to submit for each task.

0.2 Compatibility

In Task 3, our implementation of the serverless runtime observed good compatiblity with the provided serverless runtime.

1. The serverless runtime can be deployed to the Kubernetes cluster and the user module can be invoked by the serverless runtime.
2. We did not need to change the user module to make it compatible with the new serverless runtime.
3. We added more error handling, such as logging debug info and throwing errors mechanisms to the serverless runtime to make it more robust.
4. Another difference between our implementation and the provided serverless runtime is that we execute the handler function more frequently.
5. We did not need to change the dashboard code to make it compatible with the new serverless runtime. we tested both the bare and containerized version of the dashboard and they both runs well.

0.3 Dashboard

Our dashboard monitors 4 different metrics: CPUs usage, virtual memory usage, CPU frequency, and number of processes.

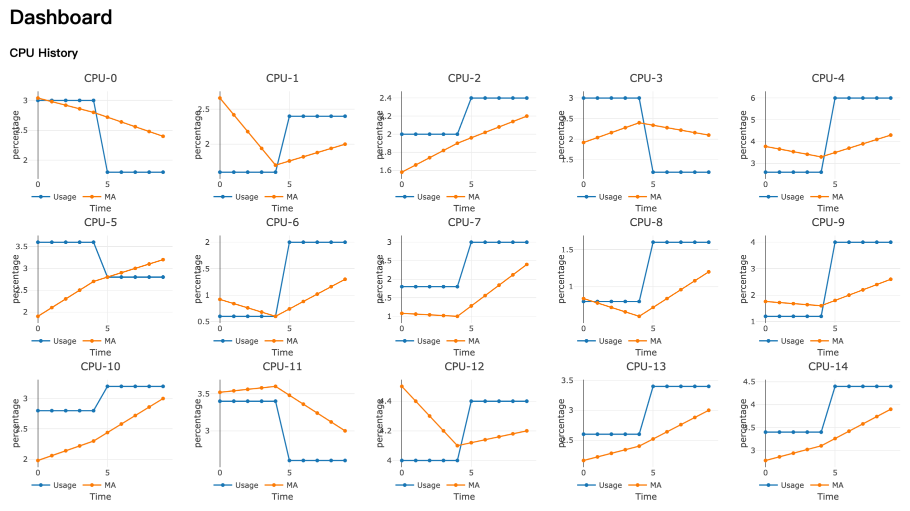

Figure 1: CPU Usage

Figure 1 shows the visualization of history and moving average of CPU usages.

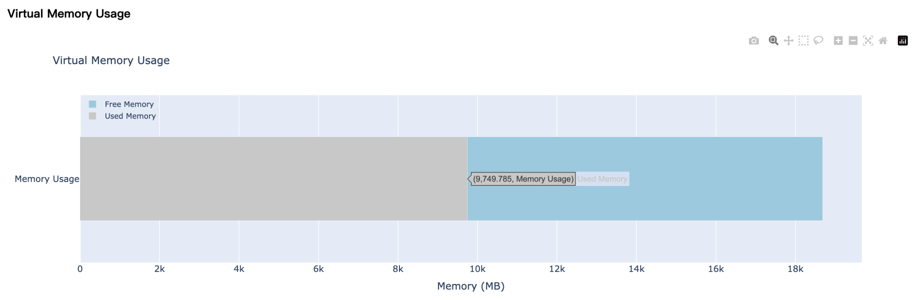

Figure 2: Virtual Memory Usage

Figure 2 shows the used and free virtual memory.

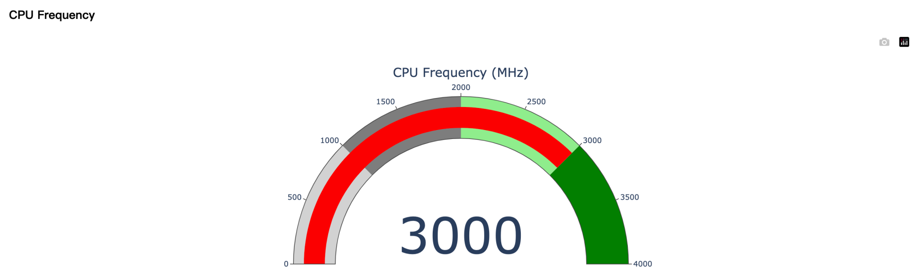

Figure 3: CPU Frequency

Figure 3 shows the current CPU speed.

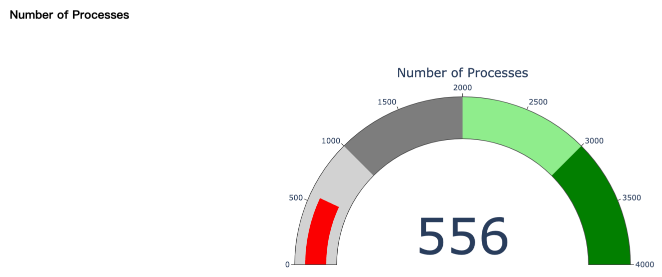

Figure 4: Number of Processes

Figure 4 shows the current number of processes.