UNIVERSITY OF MALAYA

ASSIGNMENT FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE

ACADEMIC SESSION 2023/2024 : SEMESTER 2

WIX1002 : Fundamentals of Programming

Background information:

Modern scientific collaborations have opened the opportunity of solving complex problems that involve multidisciplinary expertise and large-scale computational experiments. These experiments comprise a sequence of processing steps, or referred as tasks, that need to be executed on selected computing platforms. To support this research, UM has formed the Data Intensive Computing Centre (DICC)[1] to provide High Performance Computing (HPC) services to the campus community.

High Performance Computing is a practise of harnessing the power of aggregated computers or supercomputer, to solve complex problems that involves massive calculation and huge volume of data. It has been achieved by connecting many servers/workstations via high-speed network and managed it with a set of cluster management middleware and supporting systems.

From the user's perspective, a HPC cluster is like an everyday use computer, except it is way more powerful, as the calculation is taking place in many computing nodes and run in parallel. Users' login from their computers to the HPC cluster login node using the **SSH** program or web browser (e.g. via Open OnDemand portal), to submit their calculation, known as jobs.

There may be different types of nodes for different types of jobs. Our HPC cluster consists of the following nodes:

- **VPN Gateway**, where users establish a private connection to interact with Login Node and internal HPC resources.
- **Login Node**, where users log in via SSH or Open OnDemand portal.
- **CPU compute nodes** (where majority of computations will be executed)
- **GPU compute nodes** (for those jobs that can benefit from the massive parallel execution on Graphical Processing Unit)
- **Storage nodes** (where the data is been stored)

Figure 1 shows a brief overview of the architecture of UMHPC.

[1] Data Intensive Computing Centre (DICC): https://www.dicc.um.edu.my/

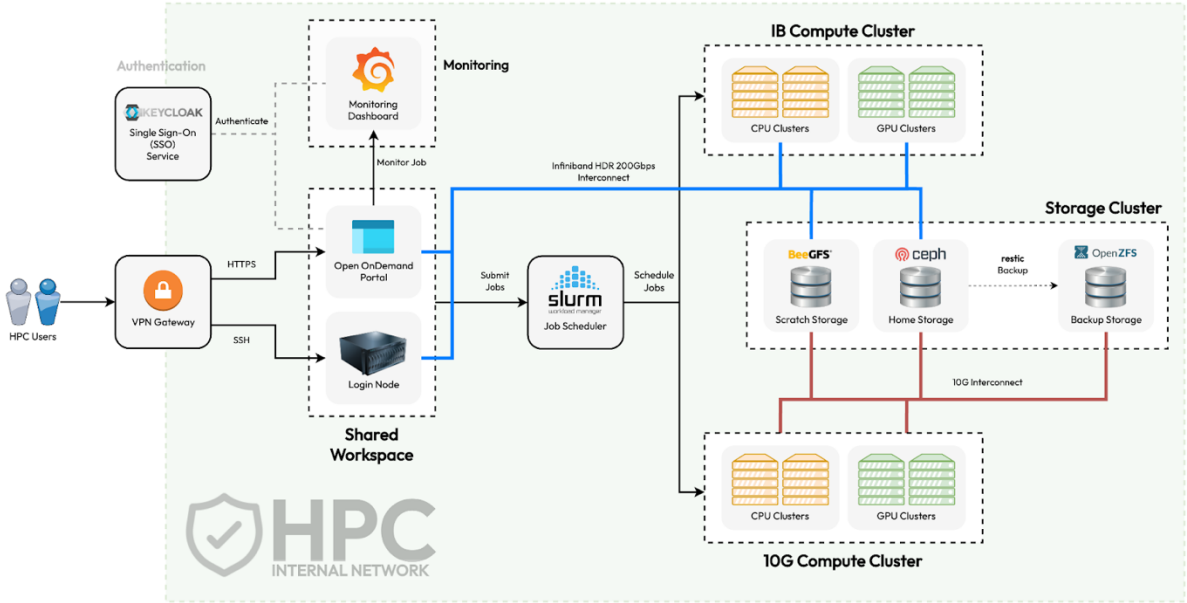

Figure 1: Architecture of UMHPC

To ensure the quality of services provided, the DICC team has deployed necessary monitoring tools to ensure the availability and stability of the system. Going through the system log files has become a daily routine for the HPC specialist in DICC for system health check. The HPC specialist will define the performance metrics and try to extract the required information from the million rows of log file. This not only require a very understanding of the HPC system, but also a very sound programming skill to make sense from these log files.

A real-world log file of the SLURM scheduler has been given in this assignment.

```
[2022-06-01T09:16:23.166] _slurm_rpc_submit_batch_job: JobId=42808 InitPrio=20861 usec=331
[2022-06-01T09:16:23.309] sched: Allocate JobId=42808 NodeList=cpu01 #CPUs=1 Partition=cpu-opteron
[2022-06-01T09:16:55.690] _slurm_rpc_submit_batch_job: JobId=42809 InitPrio=20861 usec=408
[2022-06-01T09:16:56.360] sched: Allocate JobId=42809 NodeList=cpu01 #CPUs=1 Partition=cpu-opteron
[2022-06-01T09:28:29.792] _job_complete: JobId=42808 WEXITSTATUS 0
[2022-06-01T09:28:29.793] _job_complete: JobId=42808 done
[2022-06-01T09:28:44.482] _job_complete: JobId=42809 WEXITSTATUS 0
[2022-06-01T09:28:44.482] _job_complete: JobId=42809 done
[2022-06-01T09:36:15.587] _slurm_rpc_submit_batch_job: JobId=42810 InitPrio=21292 usec=404
[2022-06-01T09:59:44.733] _slurm_rpc_submit_batch_job: JobId=42811 InitPrio=21292 usec=423
```

From the sample above,

sched: Allocate JobId=50096 NodeList=gpu03 #CPUs=16 Partition=gpu-k10

indicates a job has been started, and

_job_complete: JobId=50096 done

indicates job has ended.

Your task:

1. Study the six-months log file extracted from the SLURM controller (refer to the attached **extracted_log**).

2. Write a program to extract the useful information and show in **table** and **graph**
   (can use online tools or write code to represent in a programming way).

   useful information

   - number of job submissions
     _slurm_rpc_submit_batch_job:
   - number of job allocations
     sched: Allocate
   - number of job completions
     _job_complete: JobId=42784 done
   - number of job terminations
     Terminate signal (SIGINT or SIGTERM) received
   - number of errors happened
     error: This association
   - jobs by epyc
     Partition=cpu-epyc
   - jobs by cpu-opteron
     Partition=cpu-opteron
   - jobs by gpu
     Partition=gpuv100s, gpu-titan, gpu-k10, gpu-k40c

3. Some of the interesting metrics include (but not limited to):

   (a) Number of jobs created/ended within a given time range.

   (b) Number of jobs by partitions, i.e. EPYC, Opteron and GPU.

   (c) Number of jobs causing error and the corresponding user. The error is indicated as "[2022-06-01T15:12:23.290] error: This association…"

   (d) Average execution time of the jobs submitted to UMHPC.

(e) Other statistical data that you can extract.

- number of job terminations

Terminate signal

- number of job was killed

REQUEST_KILL_JOB

- number of jobs backfilled by scheduler

sched/backfill:

- number of jobs getting cleaned up

cleanup_completing:

- error by quality of service

error: Invalid qos

- number of times umhpc node went down

Down nodes: umhpc

4. You can refer to the detail list of attributes of SLURM here[2].

5. Discuss the outcome of your work and your finding, e.g. challenges that you faced and your solutions, in a form of a report. The report together with the source code, should be submitted to spectrum by 13th June 12:00pm.

6. Prepare for a **five minute** presentation, to share your finding and perform a system demo on week 14th, i.e. 13th to 14th June, during the lab session.

(20 marks)

[2] SLURM manual: https://slurm.schedmd.com/sacct.html

Marking scheme:

- **The report and solutions: 15 marks**
  - Requirement analysis: The metrics that you have chosen.
  - The architectural design: The source code developed to solve the problem.
  - The result and finding: The sample output of the program and the discussion on the challenges and issues you faced in developing the solutions.

- **The presentation: 5 marks**
  - A good presentation should comprise the following:
    - A good overview to set the scene for the presentation.
    - A clear explanation of the use case for the audience to know what you are trying to solve.
    - A clear architectural discussion for the audience to understand how you solve it.
    - A short but solid demonstration of the overall system.
    - An impactful closing to conclude your finding.

END