

Yeo Meng Han  
A0251772A  
EE3801 Lab 7B

Submit the following task for lab report for Part B:

### Step 26

```
Start Times
==> rplpl-slurm.queue1-dy-m5a-xlarge-1.1.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=12, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)

==> rs1a-slurm.queue1-dy-m5a-4xlarge-1.3.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=20, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)

==> rs2a-slurm.queue1-dy-m5a-4xlarge-2.4.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=20, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)

==> rs3a-slurm.queue1-dy-m5a-4xlarge-3.5.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=20, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)

==> rs4a-slurm.queue1-dy-m5a-4xlarge-1.6.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=13, tm_min=10, tm_sec=32, tm_wday=5, tm_yday=301, tm_isdst=0)

==> rse-slurm.queue1-dy-m5a-xlarge-1.2.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=12, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)
```

```
==> slurm-451.out <==
End Times
==> rplpl-slurm.queue1-dy-m5a-xlarge-1.1.out <==
Object loaded from file unity_71bf.hkl
Object loaded from file eyelink_24d5.hkl
Found path: /data/RCP/VirtualMaze.x86_64
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=37, tm_sec=54, tm_wday=5, tm_yday=301, tm_isdst=0)
1508.0303106307983

==> rs1a-slurm.queue1-dy-m5a-4xlarge-1.3.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=13, tm_min=5, tm_sec=23, tm_wday=5, tm_yday=301, tm_isdst=0)
2676.4482605457306
{
  "MessageId": "3ef88dfc-f2a0-5b20-bc28-d04bfbf6bb56"
}

==> rs2a-slurm.queue1-dy-m5a-4xlarge-2.4.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=13, tm_min=6, tm_sec=24, tm_wday=5, tm_yday=301, tm_isdst=0)
2737.4788780212402
{
  "MessageId": "8c71027b-1359-5c59-bb35-5fcc1aca2942"
}

==> rs3a-slurm.queue1-dy-m5a-4xlarge-3.5.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=13, tm_min=8, tm_sec=22, tm_wday=5, tm_yday=301, tm_isdst=0)
2856.003537416458
{
  "MessageId": "6631a09b-d828-52c9-95a2-fd2cd2210287"
}

==> rs4a-slurm.queue1-dy-m5a-4xlarge-1.6.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=13, tm_min=37, tm_sec=1, tm_wday=5, tm_yday=301, tm_isdst=0)
1589.3496520519257
{
  "MessageId": "ebd94b39-dc45-51be-be06-168ecc8b6606"
}

==> rse-slurm.queue1-dy-m5a-xlarge-1.2.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=18, tm_sec=29, tm_wday=5, tm_yday=301, tm_isdst=0)
342.5053517818451
{
  "MessageId": "190554f3-ceb5-51a8-a8c6-094530e299d1"
}
```

First 5 timestamps:

1. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=35, tm_sec=22, tm_wday=5, tm_yday=301, tm_isdst=0)`
2. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=35, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)`
3. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=35, tm_sec=57, tm_wday=5, tm_yday=301, tm_isdst=0)`
4. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=37, tm_sec=19, tm_wday=5, tm_yday=301, tm_isdst=0)`
5. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=39, tm_sec=42, tm_wday=5, tm_yday=301, tm_isdst=0)`

Last 5 timestamps:

1. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=16, tm_min=58, tm_sec=5, tm_wday=5, tm_yday=301, tm_isdst=0)`
2. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=16, tm_min=58, tm_sec=47, tm_wday=5, tm_yday=301, tm_isdst=0)`
3. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=17, tm_min=4, tm_sec=2, tm_wday=5, tm_yday=301, tm_isdst=0)`
4. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=17, tm_min=15, tm_sec=50, tm_wday=5, tm_yday=301, tm_isdst=0)`
5. `time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=17, tm_min=21, tm_sec=32, tm_wday=5, tm_yday=301, tm_isdst=0)`

### Calculating total time taken:

First start time (from `rplpl-slurm`):

`time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=12, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)`

Latest end time (from `rpllp` or `rplhighpass-sort`):

`time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=17, tm_min=21, tm_sec=32, tm_wday=5, tm_yday=301, tm_isdst=0)`

Total time taken:

Total Time Taken = (2023-10-28 17:21:32) - (2023-10-28 12:12:46)

Hours: 17 - 12 = 5 hours

Minutes: 21 - 12 = 9 minutes

Seconds: 32 - 46 = -14 seconds (Note: Negative seconds indicate borrowing from the minutes)

So, the total time taken is **5 hours, 8 minutes, and 46 seconds**.

### Step 29 (for personal reference)

- Find missing firings.mda: `find . -name "firings.mda" | cut -d "/" -f 3 > firings_channels.txt`
- To extract channels from chs.txt: `sed 's/.*\\/' chs.txt > extracted_channels.txt`
- To compare firings.txt and extract\_channels.txt: `comm -23 <(sort extracted_channels.txt) <(cut -d '/' -f 2 < firings_channels.txt | sort) > missing_channel.txt`
- Extract the missing channel's directories from chs and put it into missing-sort-chs.txt: `grep -F -f missing_channel.txt chs.txt > missing-sort-chs.txt`

### Step 30

```
cwd=`pwd`; for i in `cat missing-sort-chs.txt`; do echo $i; cd $i; sbatch /data/src/PyHipp/rplhighpass-sort-slurm.sh; cd $cwd; done
```

2018110

```
00010 PARTITION NAME USER ST TIME REASON REJECTS(REASON)
[ec2-user@ip-10-0-11-26 20181105]$ bash /data/src/PyHipp/checkfiles2.sh
Number of hkl files
665
Number of mda files
110
Start Times
==> rplpl-slurm.queue1-dy-m5a-xlarge-1.1.out <==
time.struct_time(tm_year=2023, tm_mon=10, tm_mday=28, tm_hour=12, tm_min=12, tm_sec=46, tm_wday=5, tm_yday=301, tm_isdst=0)
```

### Step 35

20181101

```
[ec2-user@ip-10-0-0-20 20181101]$ cd /7/20181101
[ec2-user@ip-10-0-0-20 20181101]$ bash /data/src/PyHipp/checkfiles2.sh
Number of hkl files
665
Number of mda files
110
Start Times
==> rplpl-slurm.queue1-dy-m5a-xlarge-1.1.out <==
```

20181102

```
}
[ec2-user@ip-10-0-0-20 20181102]$ bash /data/src/PyHipp/checkfiles2.sh
Number of hkl files
665
Number of mda files
110
Start Times
```

hkl files for each of the days:

```
(env1) [ec2-user@ip-10-0-0-20 picasso]$ nano /data/src/PyHipp/checkdays.sh
(env1) [ec2-user@ip-10-0-0-20 picasso]$ bash /data/src/PyHipp/checkdays.sh
20180702: 5
20180704: 5
20180705: 5
20180706: 5
20180710: 5
20180713: 5
20180716: 5
20180717: 5
20180718: 5
20180719: 5
20180723: 5
20180724: 5
20180725: 5
20180726: 5
20180727: 5
20180730: 5
20180731: 5
20180801: 5
20180802: 5
20180803: 5
20180810: 5
20180813: 5
20180814: 5
20180817: 5
20180823: 5
20180824: 5
20180827: 5
20180828: 5
20180829: 5
20180907: 5
20180912: 5
20180914: 5
20180919: 5
20180921: 5
20180924: 5
20180925: 5
20181004: 5
20181008: 5
20181010: 5
20181011: 5
20181016: 5
20181017: 5
20181101: 665
20181102: 665
20181105: 665
(env1) [ec2-user@ip-10-0-0-20 picasso]$
```

Repeat these steps from Lab 5:

### Step 34 (showing that you only have 1 instance and 1 volume running)

**Resources**

You are using the following Amazon EC2 resources in the Asia Pacific (Singapore) Region:

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	1
Load balancers	0	Placement groups	0	Security groups	4
Snapshots	2	Volumes	1		

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#)

[Migrate a server](#)

Note: Your instances will launch in the Asia Pacific (Singapore) Region

**Scheduled events**

**Asia Pacific (Singapore)**

**Service health**

[AWS Health Dashboard](#)

Region: Asia Pacific (Singapore)

**Zones**

Zone name	Zone ID
ap-southeast-1a	apse1-az1
ap-southeast-1b	apse1-az2
ap-southeast-1c	apse1-az3

**Account attributes**

[Default VPC](#)

vpc-016d3bd5850764ca4

**Settings**

- [Data protection and security](#)
- [Zones](#)
- [EC2 Serial Console](#)
- [Default credit specification](#)
- [Console experiments](#)

**Explore AWS**

**Save up to 90% on EC2 with Spot Instances**

Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. [Learn more](#)

**Amazon GuardDuty Malware Protection**

GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. [Learn more](#)

**Enable Best Price-Performance with AWS Graviton2**

AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad

### Step 36

**Bills**

Page refresh time: Tuesday, October 31, 2023 at 9:33:28 AM GMT+8

**AWS estimated bill summary**

Total charges and payment information

Account ID: 547215547739

Billing period: October 1 - October 31, 2023

Service provider: Amazon Web Services Singapore Private Limited

Total in USD: USD 0.00

**Estimated grand total: USD 0.00**

Payable by Account ID: 870849704511

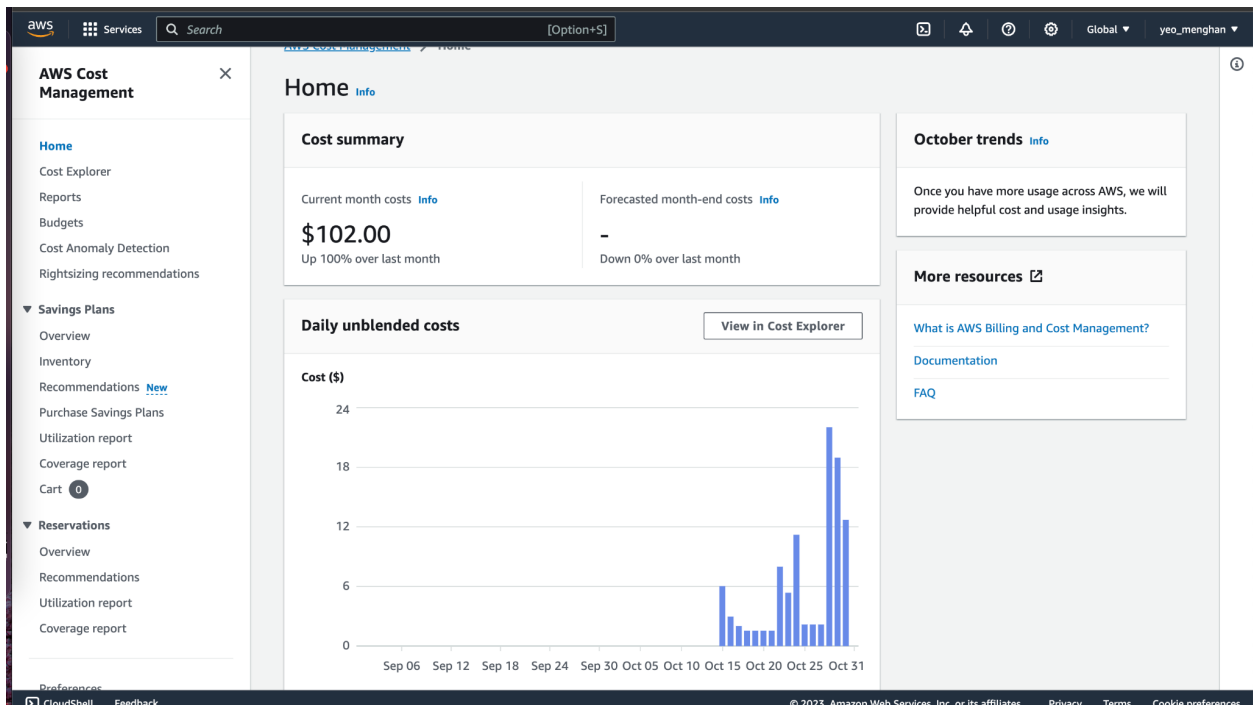
**Payment information**

**Highest estimated cost by service provider**

Viewing Amazon Web Services Singapore Private Limited

Highest service spend	Trend compared to prior month	Highest AWS Region spend	Trend compared to prior month
USD 0.00	No data to display.	USD 101.77	No data to display.

## Step 37



## Step 75

