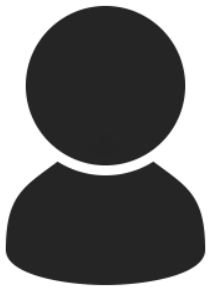
PSC Interns 2018 Final Report Presentation: Building a Data Pipeline for Calima Dashboard

Date: 2018/08/04
Intern: Alice (Chia-Hua Lee)

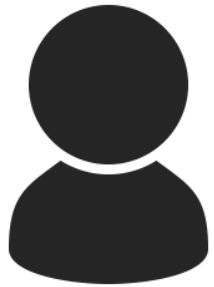
Agenda

1. Motivation
2. Data Pipeline
3. Data Source
4. Data Preprocessing
5. Dashboard Demo
6. Project Benefit for the Center

1. Motivation

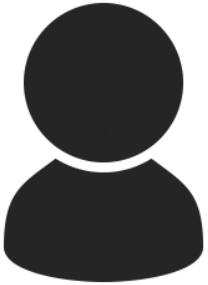


System Administrator



Consultants

Deliver the right data to the right person



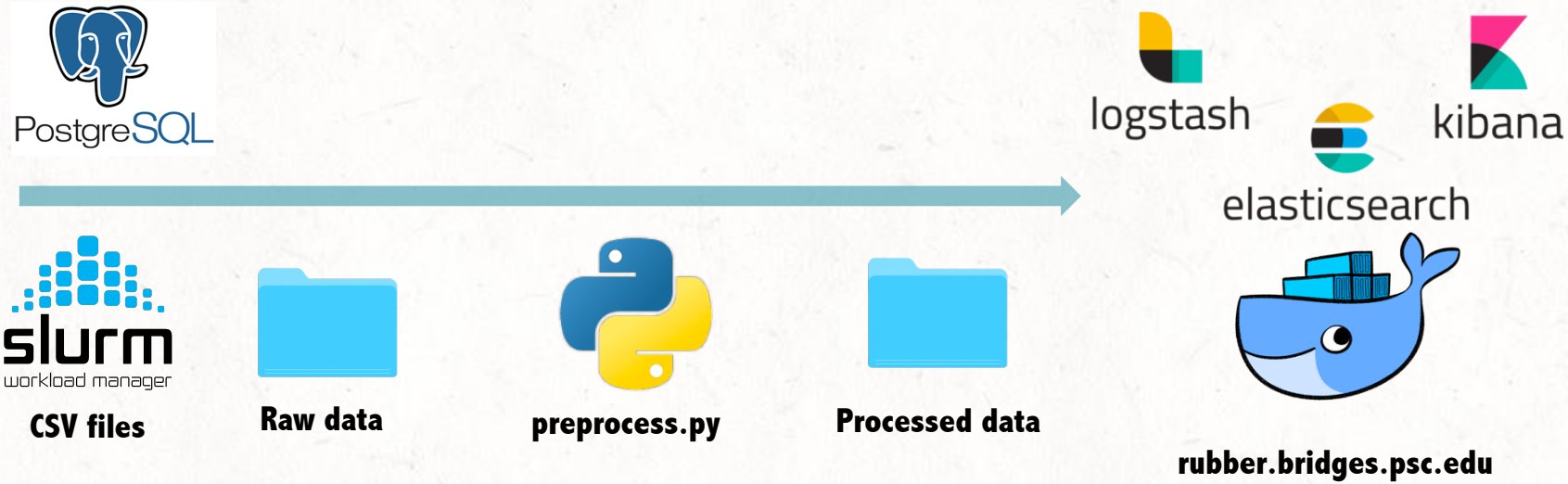
Director

• • •

End Users

2.
Data Pipeline

Calima Dashboard Data Pipeline

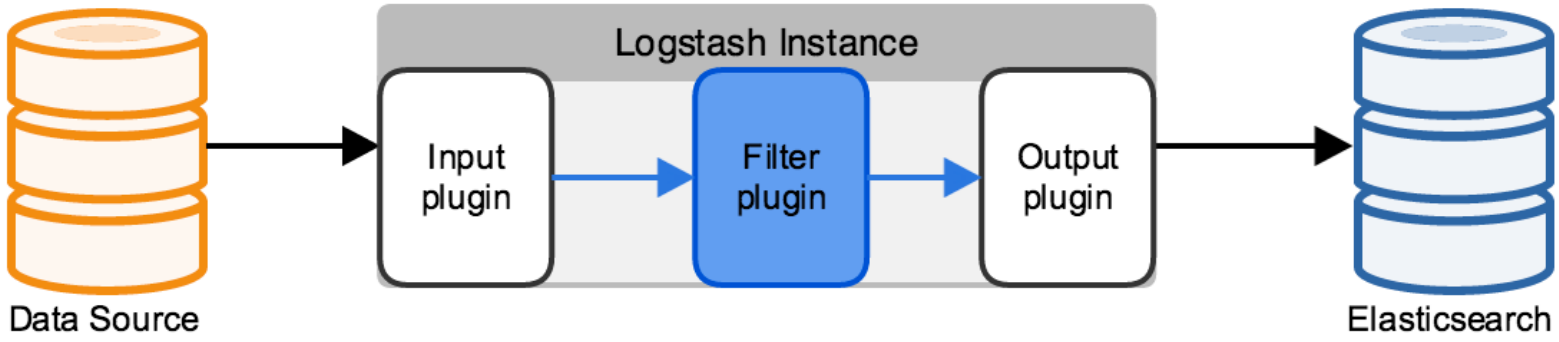



```
#!/bin/bash

function gpio()
{
    local verb=$1
    local pin=$2
    local value=$3

    local pins=($GPIO_PINS)
    if [[ "$pin" -lt ${#pins[@]} ]]; then
        local pin=${pins[$pin]}
    fi

    local gpio_path=/sys/class/gpio
    local pin_path=$gpio_path/gpio$pin
```

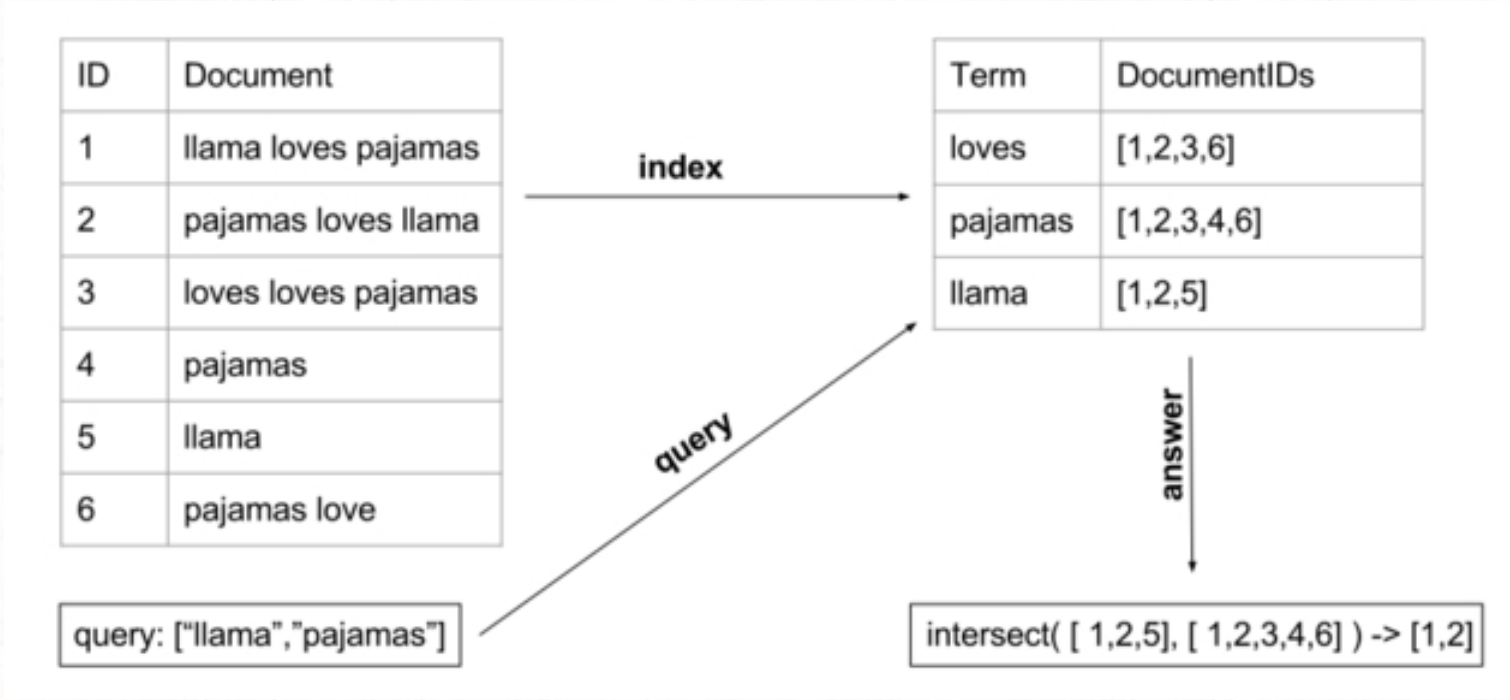
automate.sh

1. **Execute command to extract data**
2. **Invoke python script to clean data**
3. **Copy the processed data to docker (inotifywait package)**

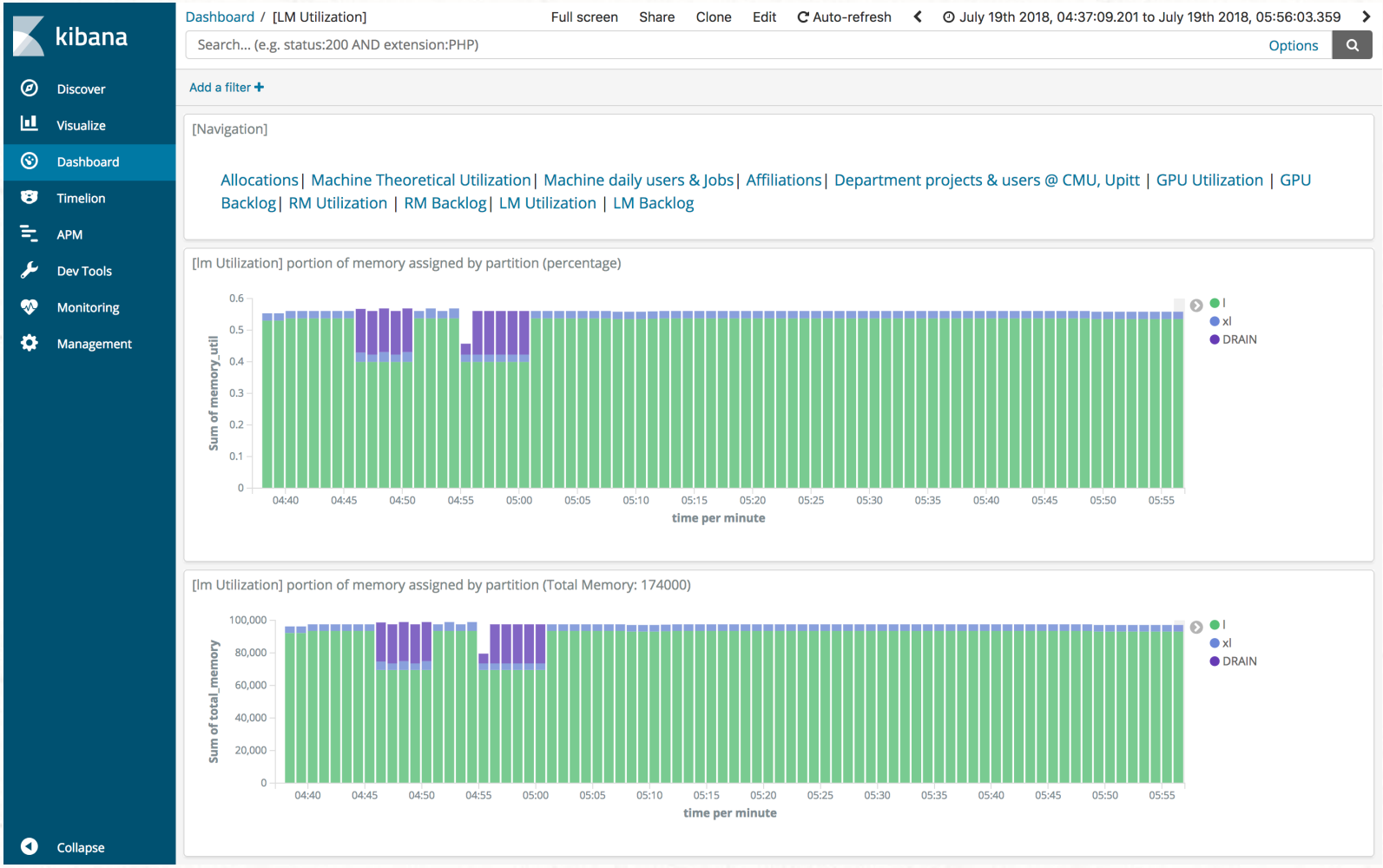
Logstash


Logstash Instance
Input plugin
Filter plugin
Output plugin
Data Source
Elasticsearch

Elasticsearch under the hood - Lucene



Kibana


kibana
Discover
Visualize
Dashboard
Timelion
APM
Dev Tools
Monitoring
Management
Collapse
Dashboard / [LM Utilization]
Full screen
Share
Clone
Edit
Auto-refresh
July 19th 2018, 04:37:09.201 to July 19th 2018, 05:56:03.359
Search... (e.g. status:200 AND extension:PHP)
Options
Add a filter +
[Navigation]
Allocations | Machine Theoretical Utilization | Machine daily users & Jobs | Affiliations | Department projects & users @ CMU, Upitt | GPU Utilization | GPU Backlog | RM Utilization | RM Backlog | LM Utilization | LM Backlog
[lm Utilization] portion of memory assigned by partition (percentage)
Sum of memory_util
0.6
0.5
0.4
0.3
0.2
0.1
0
04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55
time per minute
l
xl
DRAIN
[lm Utilization] portion of memory assigned by partition (Total Memory: 174000)
Sum of total_memory
100,000
80,000
60,000
40,000
20,000
0
04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55
time per minute
l
xl
DRAIN


3.
Data Source

Data Source: Allocations

About Allocations Database

- Development Environment: bell
- Production Environment: noll
- Data History: 2016/03/07 ~ now
- Size of View on noll Databse: 8,883,000+ rows

#	Table Name	Brief Description
1	public.subgrant	grant info
2	Public. subgrantplatinfo	platform info: Reg, GPU, Large,SU allocations
3	public.subgrantmach	machine Reg, GPU, Large
4	platmachuser	grant/platform/machine/username
5	users	grant/platform/machine/username
6	nsfaffiliations	affiliations table (eg. CMU, Pitt, USC, etc..)
7	public.jobs	subset of slurmdb focusing on jobs for accounting purposes
8	public.prorated_jobs	Truncate jobs into daily basis
9	Public.actions	new grant, renewal, supplement, time extension, SU amount

What is Allocations Database All About?

PI

Affiliation: PSC
pipscid: nystrom
Email: nick@psc.edu
Telephone: 412567983

Consultant

Subgrant Number	CDA090001P	MCB170031P	DMR160034P
Allocation	RM 20000 GPU 30000 LM 40000	RM 10000	RM 20000
Start - End	2017/07/08 -2019/04/04		
Charge ID	pscstaff	mr5612p	ca8r6hp
Title	PSC Staff	Bridges Workshop	AI & BD Intern

UserID
User Name
JobID/SU charged/Machine

Allocations Relevant Dashboard Covers the following topics :

- Allocations
- Machine Theoretical Utilization
- Machine daily users & Jobs
- Historical Resource Utilization on Different Machines by Affiliations
- Resources Utilization by Departments @ CMU & Upitt

Dashboard Link: http://rubber.bridges.psc.edu:5601/

Data Source: Slurm

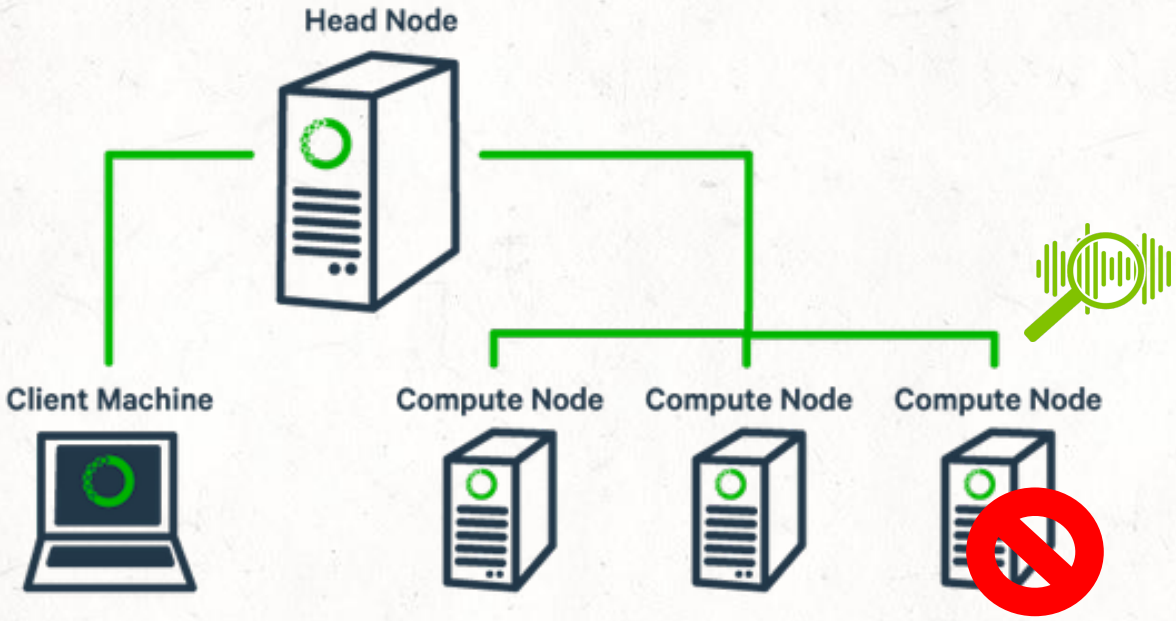
About Slurm Database

- Development/Production Environment:

  use raw CSV files directly from sacct & sacctmgr commands
- Data History of sacctmgr : since June 2018

#	Columns used for sacct	Columns used for sacctmgr
1	Account	NodeName
2	AllocTRES	TimeStart
3	JobName	TimeEnd
4	JobID	State
5	User	User
6	Partition	
7	Start	
8	State	
9	Submit	
10	ReqMem	
11	Timelimit	
12	NodeList	

What is sacct & sacctmgr Data All About?



Partition : GPU
ReqTime: 2hr
Node: 4
NodeList: gpu[036,038-040]
Submit Time: 2017/08/01 13:00:02

Get Events on down or drain nodes on clusters

Analysis on Resources Utilization & Backlog

- GPU Utilization & Backlogs

EX: Run a GPU job on 4 P100 nodes for 30 minutes is

→interact -p GPU --gres=gpu:p100:2 -N 4 –t 30:00:00

- RM Utilization & Backlogs

EX: Run in the RM-shared partition using 4 cores for 1 hour .

→interact -p RM-shared --ntasks-per-node=4 –t 1:00:00

EX: Run in the RM-shared partition using 2 Nodes for 30 minutes.

→interact -p RM -N 2 -t 30:00

- LM Utilization & Backlogs

EX: Run in the LM partition for 10 hours of wall time and 6TB of memory :

→interact -p LM - t 10:00:00 --mem 6000GB

4. Data Preprocessing (Slurm Logs)

Data Preprocessing (1) – Job Transform

Take RM Node as an example

AllocTRES **cpu=32,mem=123200M,node=1,billing/gpu=32,gres/gpu=2,gres/gpu:p100=2**

```
Account                                              tr561fp
AllocTRES     cpu=84,mem=369600M,node=3,billing/gpu=84
JobName                                       CP2K_example
JobID                                              3597006
User                                                llzhao
Partition                                               RM
Start                                  2018-07-18T12:45:12
State                                               FAILED
Submit                                 2018-07-18T10:14:15
ReqMem                                             123200Mn
Timelimit                                       2-00:00:00
NodeList                                     r[107,641,644]
End                                    2018-07-20T00:44:09
Name: 253, dtype: object
```

```
Account                                              tr561fp
JobName                                       CP2K_example
JobID                                              3597006
User                                                llzhao
Partition                                               RM
Start                                  2018-07-18T12:45:12
State                                               FAILED
Submit                                 2018-07-18T10:14:15
ReqMem                                             123200Mn
NodeList                                     r[107,641,644]
End                                    2018-07-20T00:44:09
Alloc_CPU                                               84
Alloc_MEM                                           369600
Alloc_NODE                                               3
nodeArray                                  [107, 641, 644]
ReqTime                                             172800
Name: 150, dtype: object
```

NodeList r[107, 108-110, 131, 565]

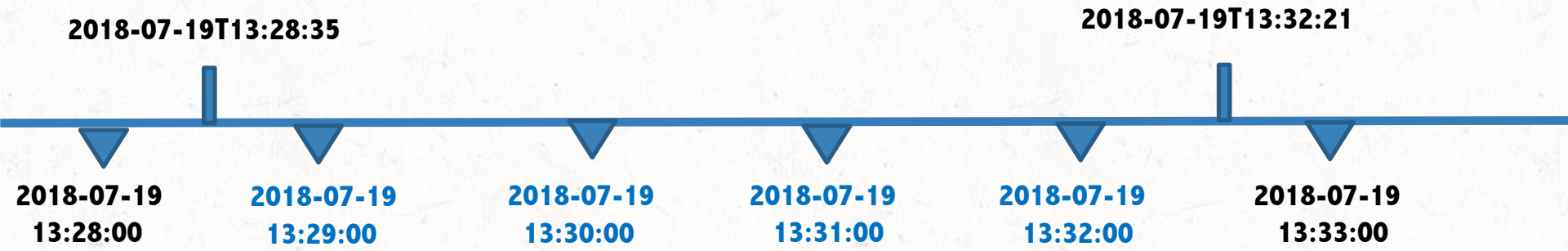
NodeList l[003, 004-005, 009],xl[001, 002]

Data Preprocessing (2) - Time

(submit) (start)

Transform start and end time to minute per row (like taking snapshot)

- If a job has a start time of 2018-07-19T13:28:35 and end time of 2018-07-19T13:32:21 in sacct, this single job will be converted to four rows with timestamp like the following:



***A job that run/waits for 10 days to be executed would have 60*24*10= 14400 rows

Preprocessed Data (August 1st –August 3rd)

GPU

Utilization

```
gpu_util_graph_data.shape (91181, 6)
```

	time	gpu	type	partition	state	Node
0	2018-07-30T00:25:00	1.0	p100	GPU-shared	TIMEOUT	1
1	2018-07-30T00:26:00	1.0	p100	GPU-shared	TIMEOUT	1
2	2018-07-30T00:27:00	1.0	p100	GPU-shared	TIMEOUT	1

Backlog

Original

(2374, 18)

```
gpu_backlog_graph_data.shape (385083, 6)
```

	time	partition	nodeType	user	reqTime	jobid
0	2018-07-18T17:36:58	GPU	k80	monje	86400.0	1
1	2018-07-18T17:37:58	GPU	k80	monje	86400.0	1
2	2018-07-18T17:38:58	GPU	k80	monje	86400.0	1

Interactive

```
gpu_interact_graph_data.shape (172, 5)
```

	Submit	Partition	nodeType	Alloc_NODE	Waittime
0	2018-07-31T22:59:32.000	GPU-shared	p100	1	66.0
1	2018-07-31T23:13:36.000	GPU-shared	k80	1	0.0
2	2018-07-31T23:31:04.000	GPU-small	p100	1	0.0

Preprocessed Data (August 1st –August 3rd)

RM

Original

(8306, 18)

Utilization

```
rm_util_graph_data.shape (94319, 5)
```

	time	core	partition	state	Node
0	2018-07-30T02:20:00	28.0	RM	TIMEOUT	4
1	2018-07-30T02:21:00	28.0	RM	TIMEOUT	4
2	2018-07-30T02:22:00	28.0	RM	TIMEOUT	4

Backlog

```
rm_backlog_graph_data.shape (1486986, 5)
```

	time	partition	user	reqTime	jobid
0	2018-07-25T01:10:32	RM	ahazel3	172800.0	1
1	2018-07-25T01:11:32	RM	ahazel3	172800.0	1
2	2018-07-25T01:12:32	RM	ahazel3	172800.0	1

Interactive

```
rm_interact_graph_data.shape (211, 4)
```

	Submit	Partition	Alloc_NODE	Waittime
0	2018-07-31T23:28:19.000	RM-small	1	0.0
1	2018-07-31T23:47:05.000	RM-small	2	0.0
2	2018-08-01T00:02:20.000	RM-small	1	0.0

Preprocessed Data (August 1st –August 3rd)

LM

Utilization

```
lm_util_graph_data.shape (108346, 6)
```

	time	mem	nodeType	state	mixNode	Node
0	2018-07-19T11:59:00	3000.0	l	TIMEOUT	l	1
1	2018-07-19T12:00:00	3000.0	l	TIMEOUT	l	1
2	2018-07-19T12:01:00	3000.0	l	TIMEOUT	l	1

Backlog

```
lm_backlog_graph_data.shape (41991, 7)
```

	time	nodeType	mixNode	user	mem	reqTime	jobid
0	2018-07-19T10:56:22	l041	abyss.script	rsleith	3000	1206000.0	1
1	2018-07-19T10:57:22	l041	abyss.script	rsleith	3000	1206000.0	1
2	2018-07-19T10:58:22	l041	abyss.script	rsleith	3000	1206000.0	1

Original

(4170, 18)

Interactive

```
lm_interact_graph_data.shape (17, 6)
```

	Submit	nodeType	Mix_Node	Alloc_MEM	Alloc_NODE	Waittime
0	2018-08-01T11:25:26.000	l	l	1	1	0.0
1	2018-08-01T13:58:59.000	l	l	500	1	0.0
2	2018-08-01T14:54:39.000	l	l	1	1	542.0

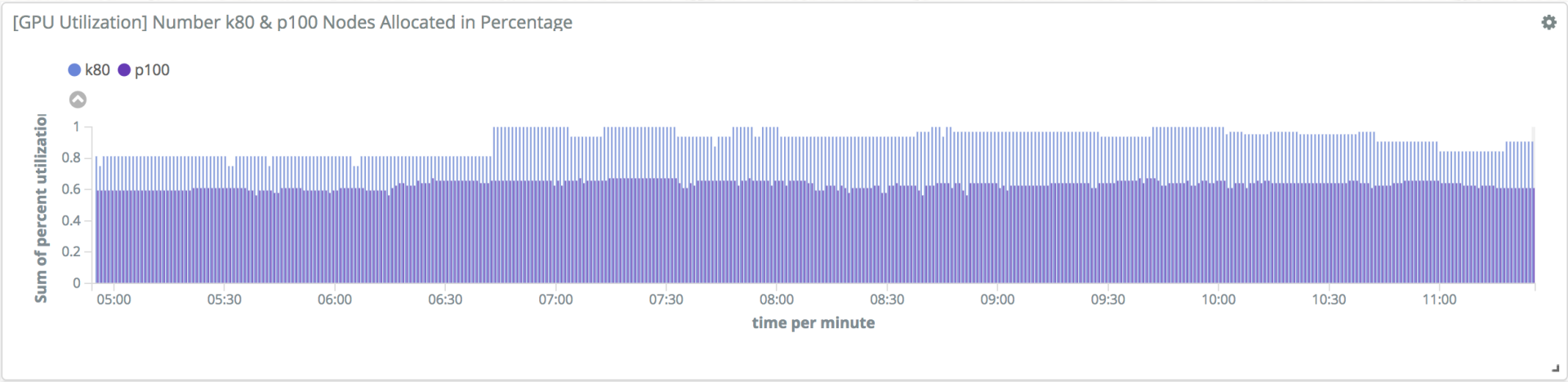
5. Dashboard Demo

GPU Utilization

Time Spam: 2018-07-19 04:29:00 – 2018-07-20 11:30

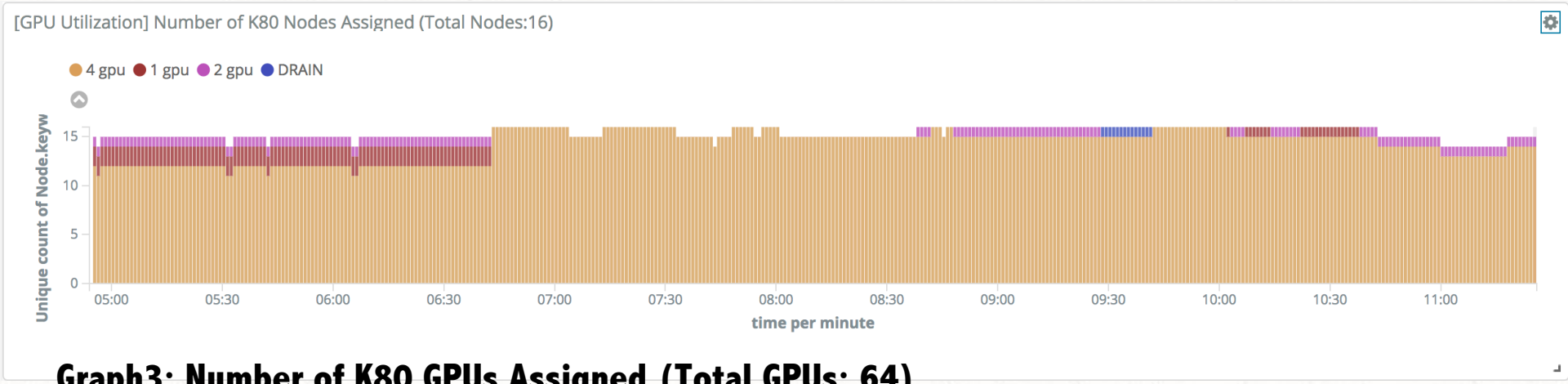
Graph1: Percentage of GPUs Assigned by Node Type

K80 & P100 both have 64 GPUs

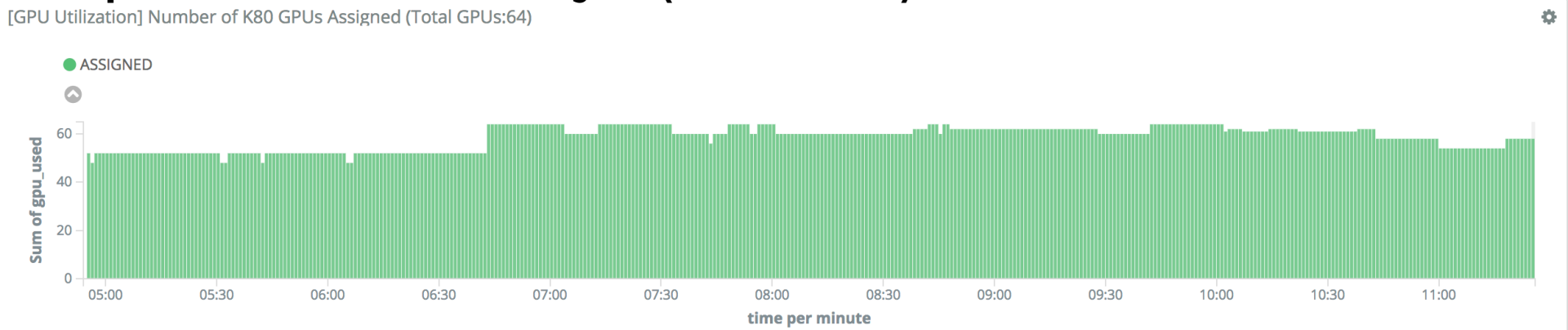


GPU Utilization

Graph2: Number of K80 Nodes Assigned (Total Nodes: 16)



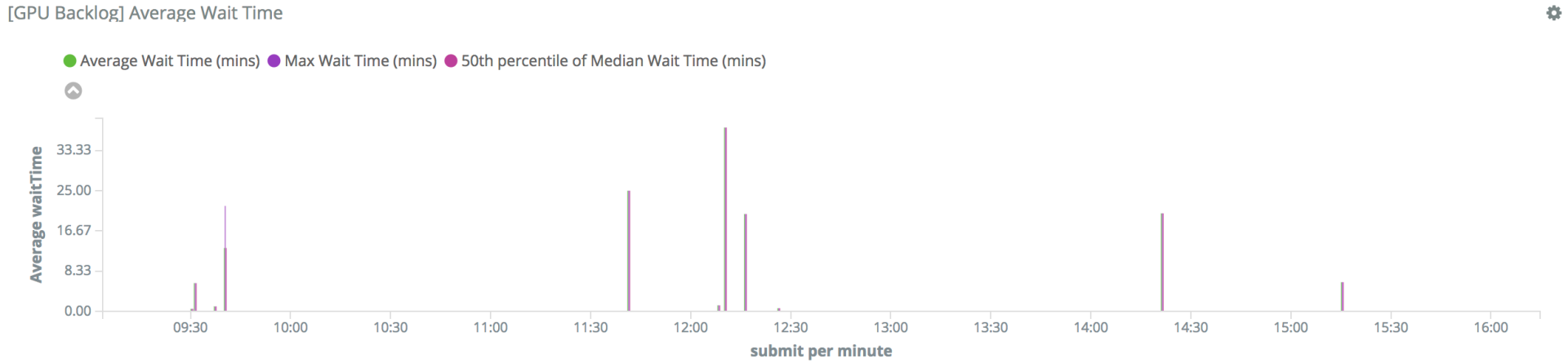
Graph3: Number of K80 GPUs Assigned (Total GPUs: 64)



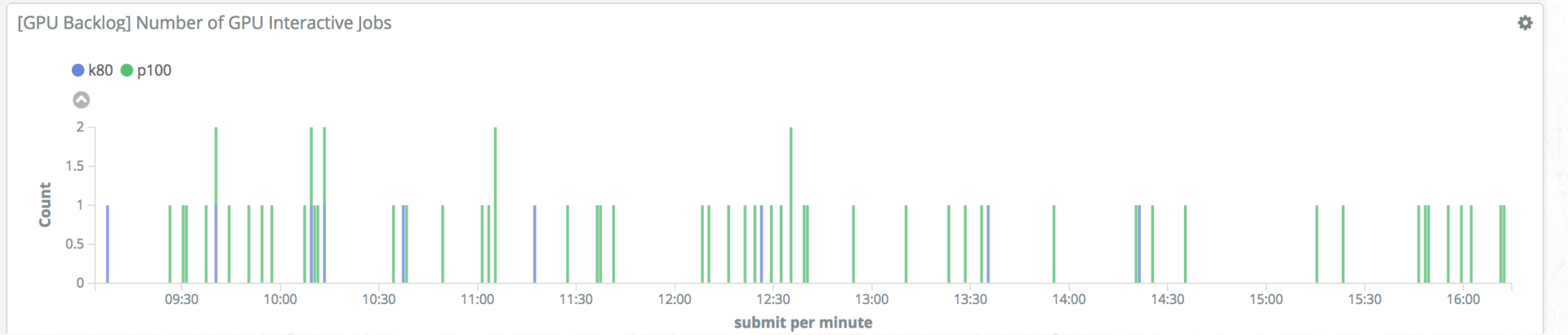
*** Drain Nodes: no further job will be scheduled on that node, but the currently running jobs will keep running

GPU Backlog

Graph 4: Average/Median/Max Wait Time for Interactive Jobs

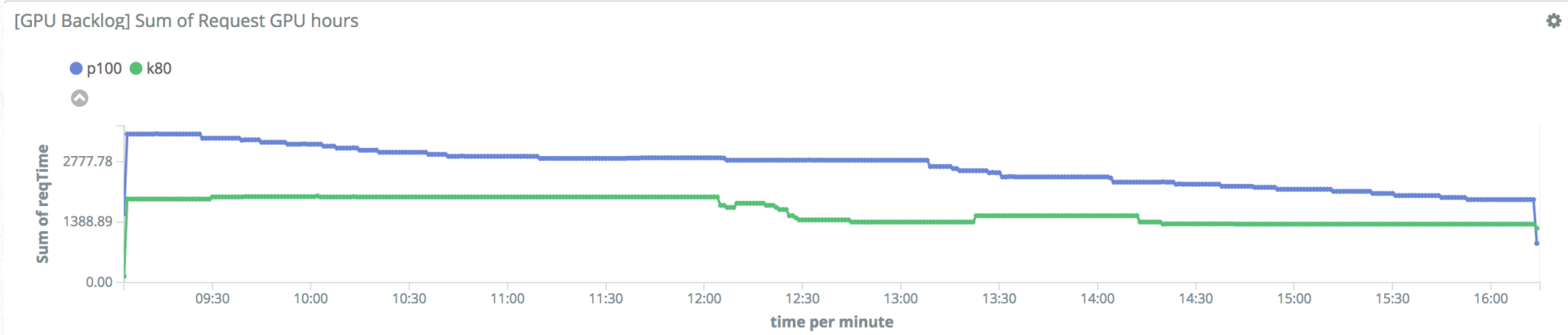


Graph 5: Number of GPU Interactive Jobs by Node Type

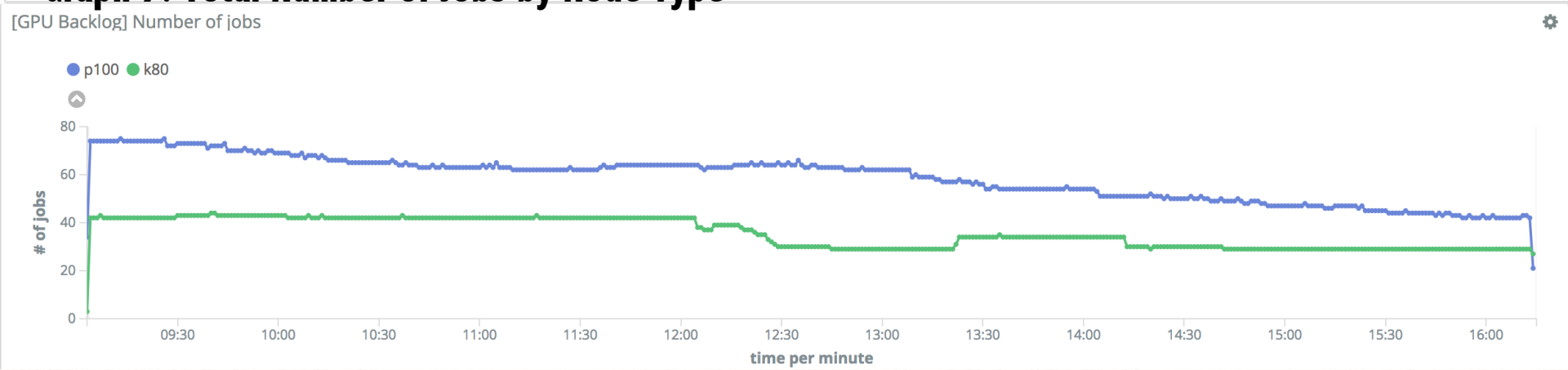


GPU Backlog

Graph 6: Sum of Request GPU Hours by Node Type

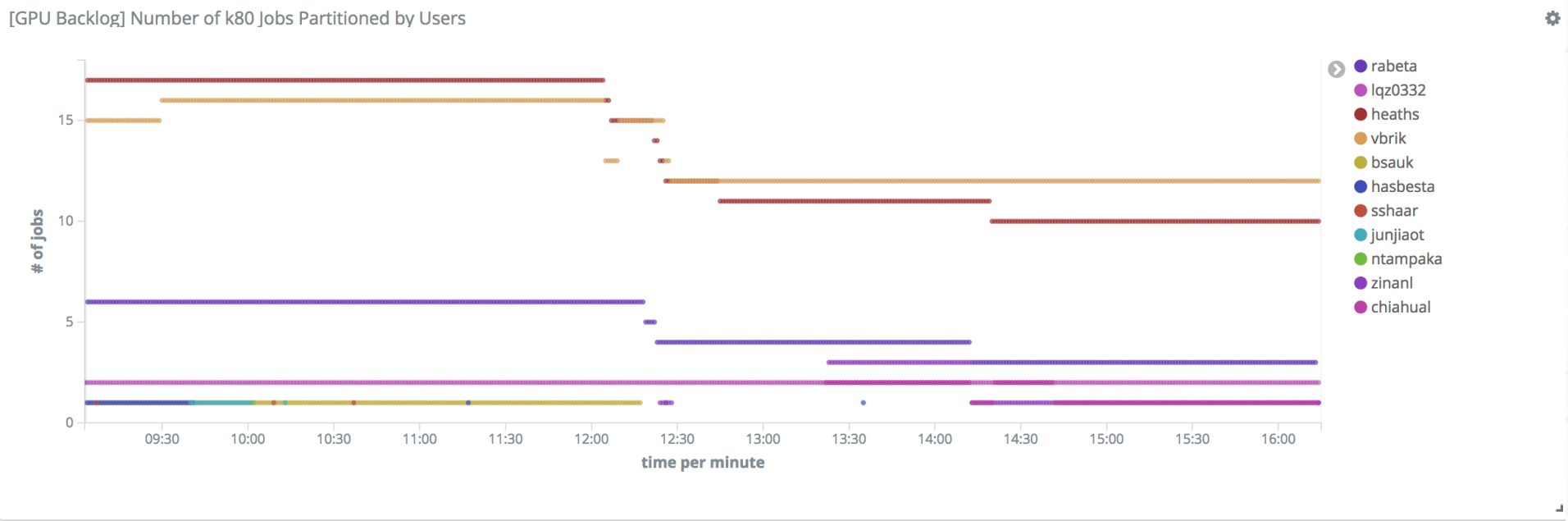


Graph 7: Total Number of Jobs by Node Type



GPU Backlog

Graph 8: Total Number of Jobs by user



6.
Project Benefits for PSC

“

- **Monitor the Utilization & Backlog of different resources on Bridges in longer time span, not just snapshot**
- **Manage Allocations more efficiently**

Thanks!

You can find me at

chiahual@andrew.cmu.edu