

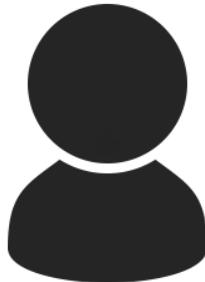
PSC Interns 2018 Final Report Presentation: Building a Data Pipeline for Calima Dashboard

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

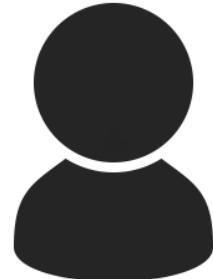
Agenda

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

1. Motivation



System
Administrator



Consultants

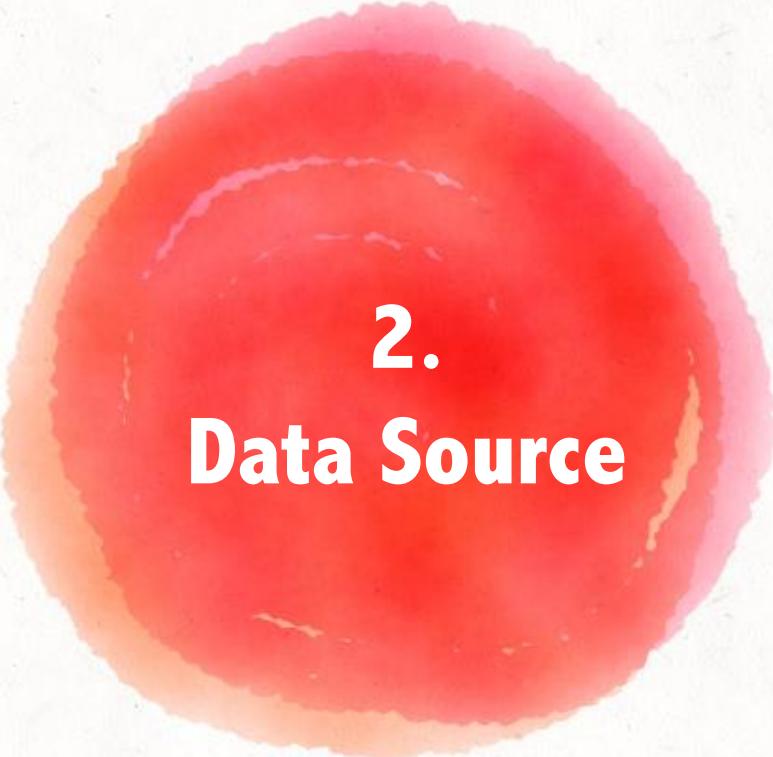


Director

**Deliver
the right data
to the
right person**



End Users



2. **Data Source**

Data Source: Allocations



About Allocations Database

- ▶ **Development Environment:** bell
- ▶ **Production Environment:** noll
- ▶ **Data History:** 2016 /03 ~ 2018/8/4
- ▶ **Data Size:** 8883,000+ 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, MIT, 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

Data Source: Allocations

users	platmachuser	subgrant	subgrantmach	nsfaffiliations	jobs	prorated_jobs
pscid	subgrantnumber	subgrantnumber	chargeid	affilcode	job_id	job_id
firstname	machine	pipscid	machine	affiliation	machine	username
lastname	username	consultant	startdate	lat	walltime	machine
department		title	enddate	long	charge_id	charge_id
affilcode			allocation		partition	job_name
telephone					start_time	prorated_jobs
email					submit_time	short_date
country						start_time
						prorated_su_charged

What is Allocations Database All About?



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

User ID
User Name
JobID/SU charged/Machine



Part 1 Dashboard Covers the following topics :

- ▶ **Allocations**
- ▶ **Machine Theoretical Utilization**
- ▶ **Machine daily users & Jobs**
- ▶ **Affiliations**
- ▶ **Department Projects & Users @ CMU, Upitt**

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

Data Source: Slurm

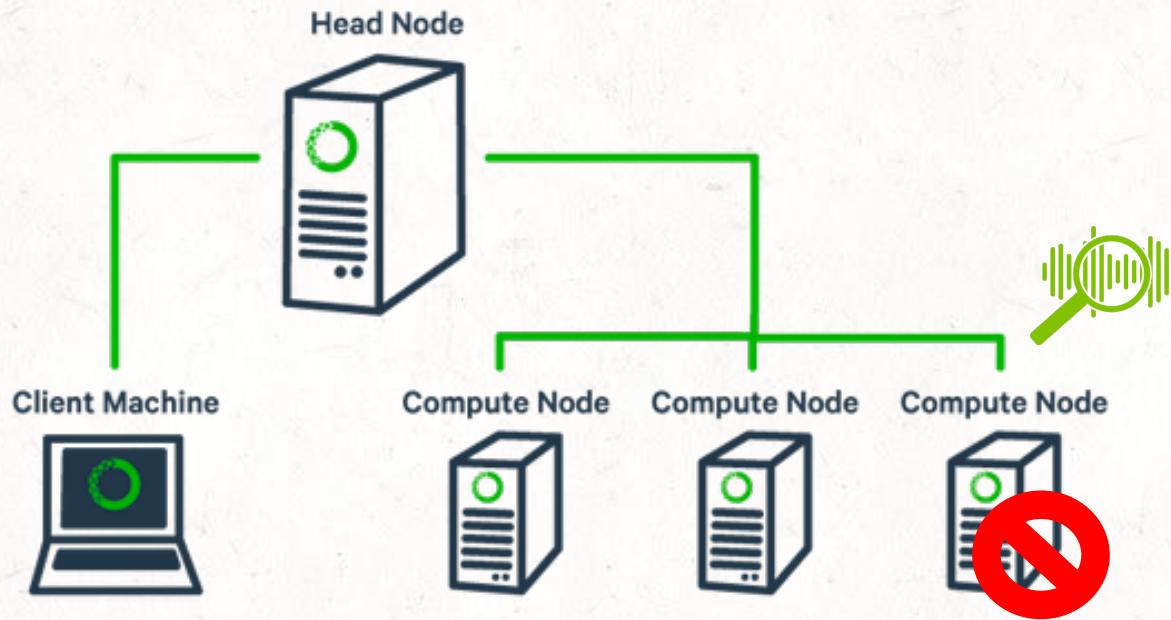


About Allocations Database

- ▶ **Development/Production Environment:**
raw CSV files directly from sacct & sacctmgr
- ▶ **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

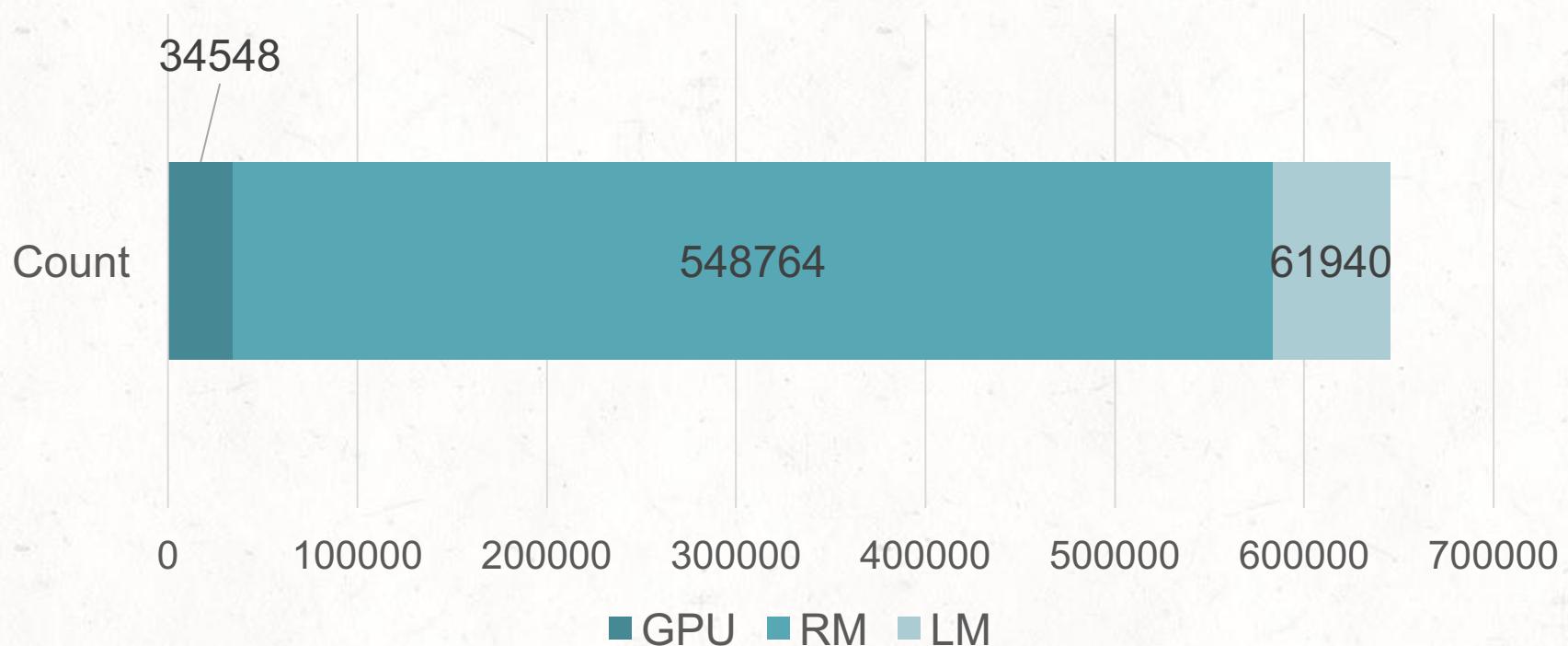
NodeList: gpu[036,038-040]

Submit Time: 2017/08/01 13:00:02

Get Events on down or drain nodes on clusters

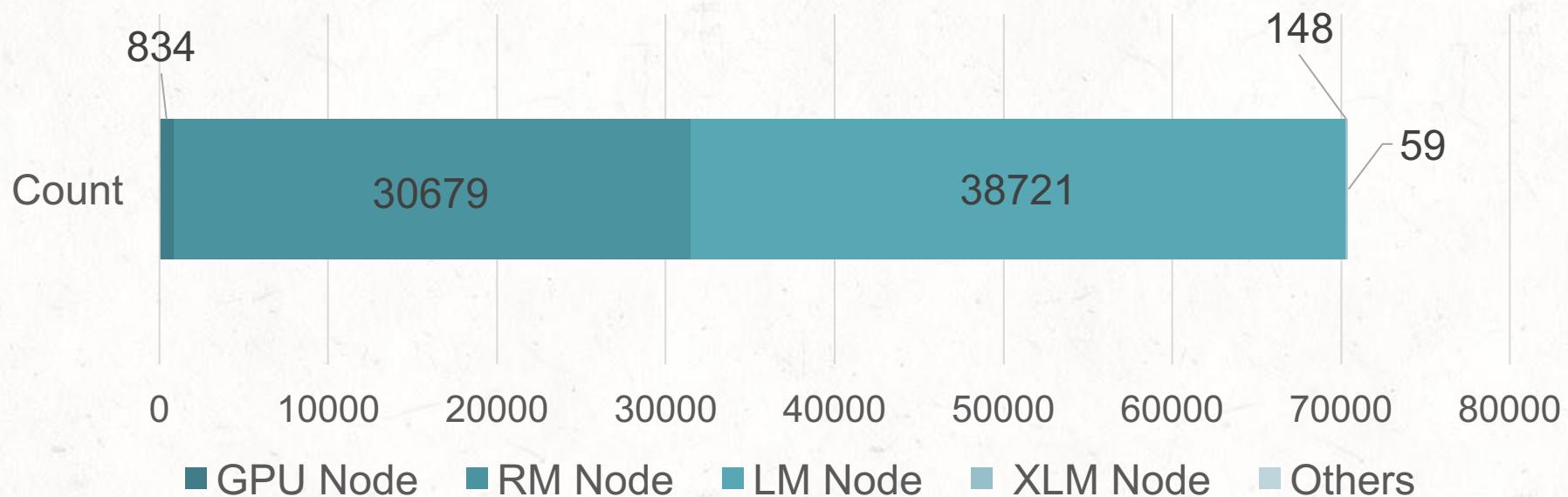
Sacct Data Size

Total Number of Jobs (June 1st ~ August 4th)



Sacctmgr Data Size

Number of Events on Nodes in Drain or Down State
(June 1st ~ August 4th)



saact & sacctmgr Raw Data

Sample sacct command for GPU nodes:

```
sacct -a -P -X -S 071918-00:00:00 -E 072018-23:59:59 --
```

```
format=Account,AllocTRES,JobName,JobID,User,Partition,Start,State,Submit,ReqMem,Timelimit,NodeList,End -s  
CANCELLED,TIMEOUT,FAILED,COMPLETED,NODE_FAIL,OUT_OF_MEMORY -r GPU,GPU-shared,GPU-small> today_gpu.csv
```

```
Account|AllocTRES|JobName|JobID|User|Partition|Start|State|Submit|ReqMem|Timelimit|NodeList|End  
ct560tp|cpu=128,mem=492800M,node=4,billing/gpu=128,gres/gpu=8,gres/gpu:p100=8|0.01%.sh|3542846|jqwang|GPU|2018-07-18T04:03:46|COMPLETED|2018-07-09T13:25:45|123200Mn|2-00:00:00|gpu[024,030,036,043]|2018-07-19T07:56:19  
ct560tp|cpu=128,mem=492800M,node=4,billing/gpu=128,gres/gpu=8,gres/gpu:p100=8|0.01%.sh|3542847|jqwang|GPU|2018-07-18T20:39:03|COMPLETED|2018-07-09T13:25:48|123200Mn|2-00:00:00|gpu[023,025-027]|2018-07-20T00:37:22  
ct560tp|cpu=128,mem=492800M,node=4,billing/gpu=128,gres/gpu=8,gres/gpu:p100=8|0.01%.sh|3542849|jqwang|GPU|2018-07-19T07:56:35|COMPLETED|2018-07-09T13:25:57|123200Mn|2-00:00:00|gpu[024,030,036,043]|2018-07-20T12:36:33
```

Sample sacctmgr command:[1](#)

```
sacctmgr show events -P event=node Start=071918-00:00:00 End=072018-23:59:59 > today_state.csv
```

```
Cluster|NodeName|TimeStart|TimeEnd|State|Reason|User  
bridges|gpu004|2018-07-19T13:27:33|2018-07-19T13:27:38|DOWN|prolog: default gw setup failed|root(0)  
bridges|gpu004|2018-07-19T13:27:38|2018-07-19T13:41:47|DRAIN|prolog: default gw setup failed|slurm(1010)  
bridges|1007|2018-07-20T18:12:34|2018-07-20T18:14:55|DOWN*|Not responding|slurm(1010)
```

Analysis on Resources

- ▶ 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 and 1 hour of walltime.

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

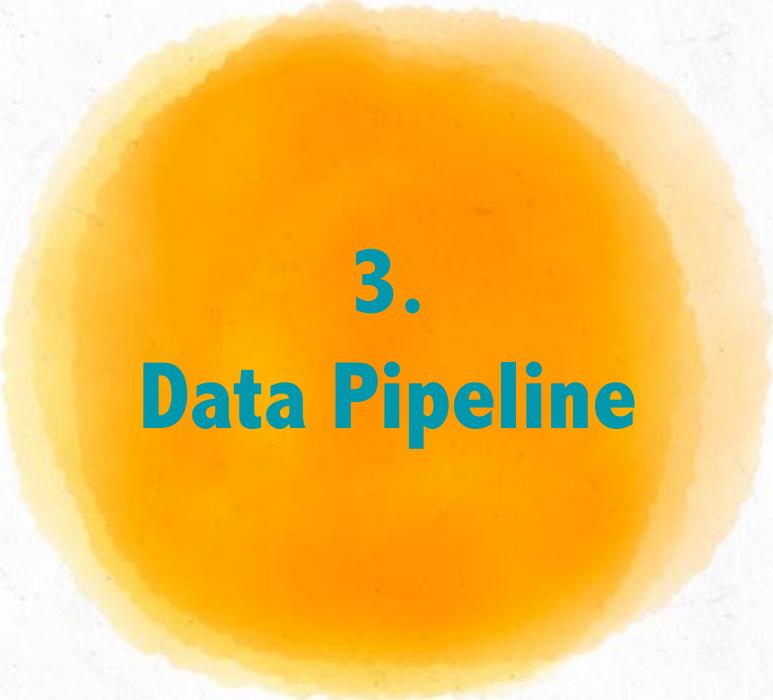
→**interact -p RM -N 2 -t 30:00**

- ▶ LM Utilization & Backlogs

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

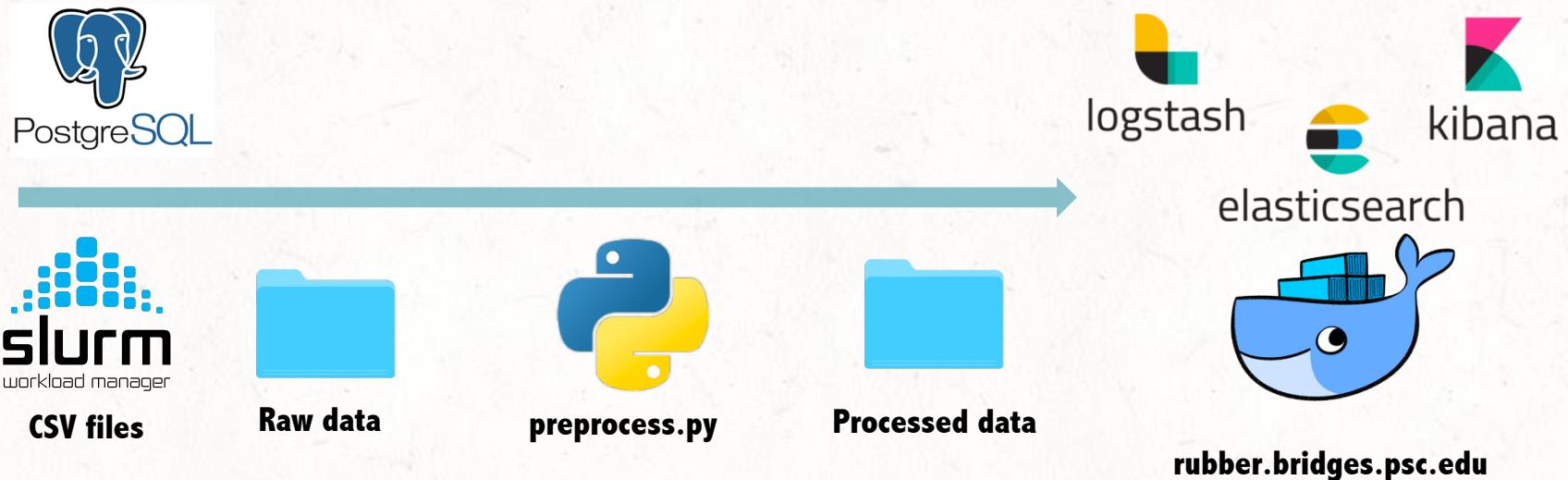
→**interact -p LM -t 10:00:00 --mem 6000GB**

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



3. **Data Pipeline**

Calima Dashboard Data Pipeline



```
#!/bin/bash

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

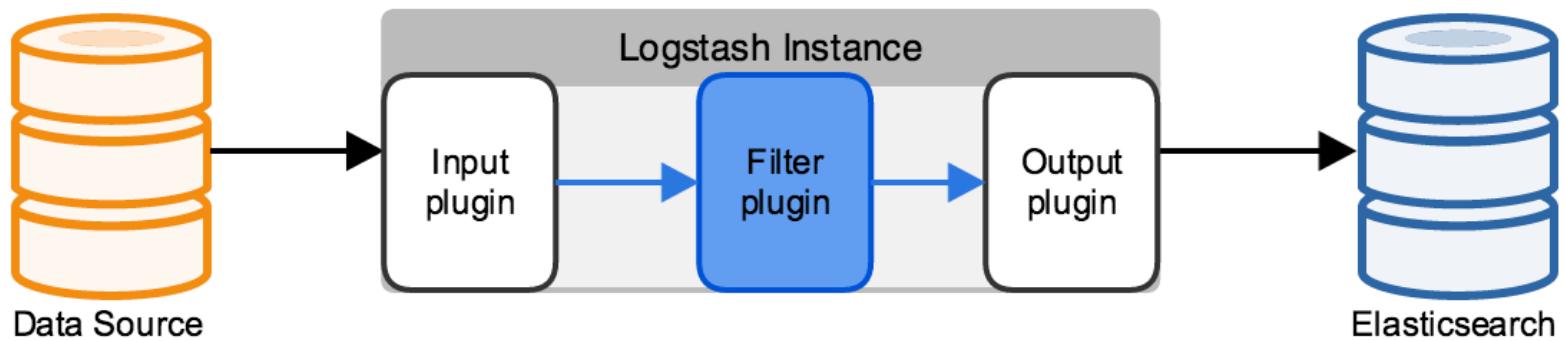
    local pins=($GPIO_PINS)
    if [[ "$pin" -lt ${#pins[@]} ]]; then
        local pin=${pins[$pin]}
    fi

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

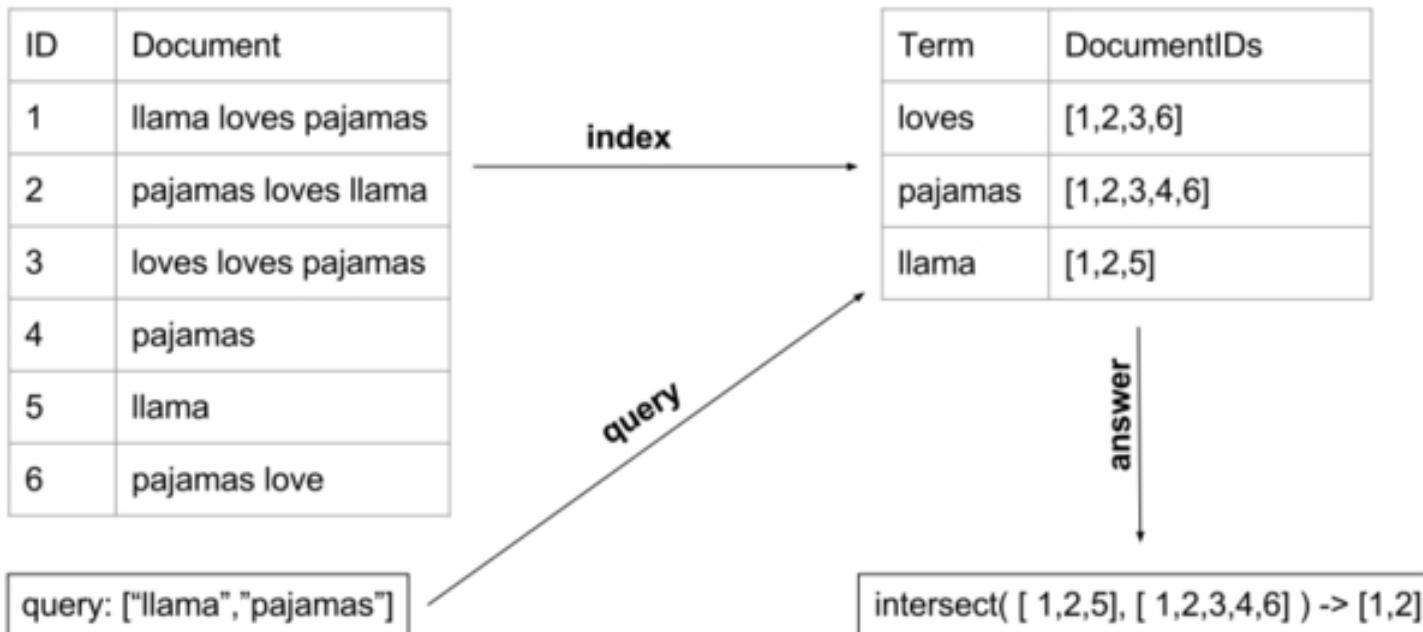
automate.sh

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

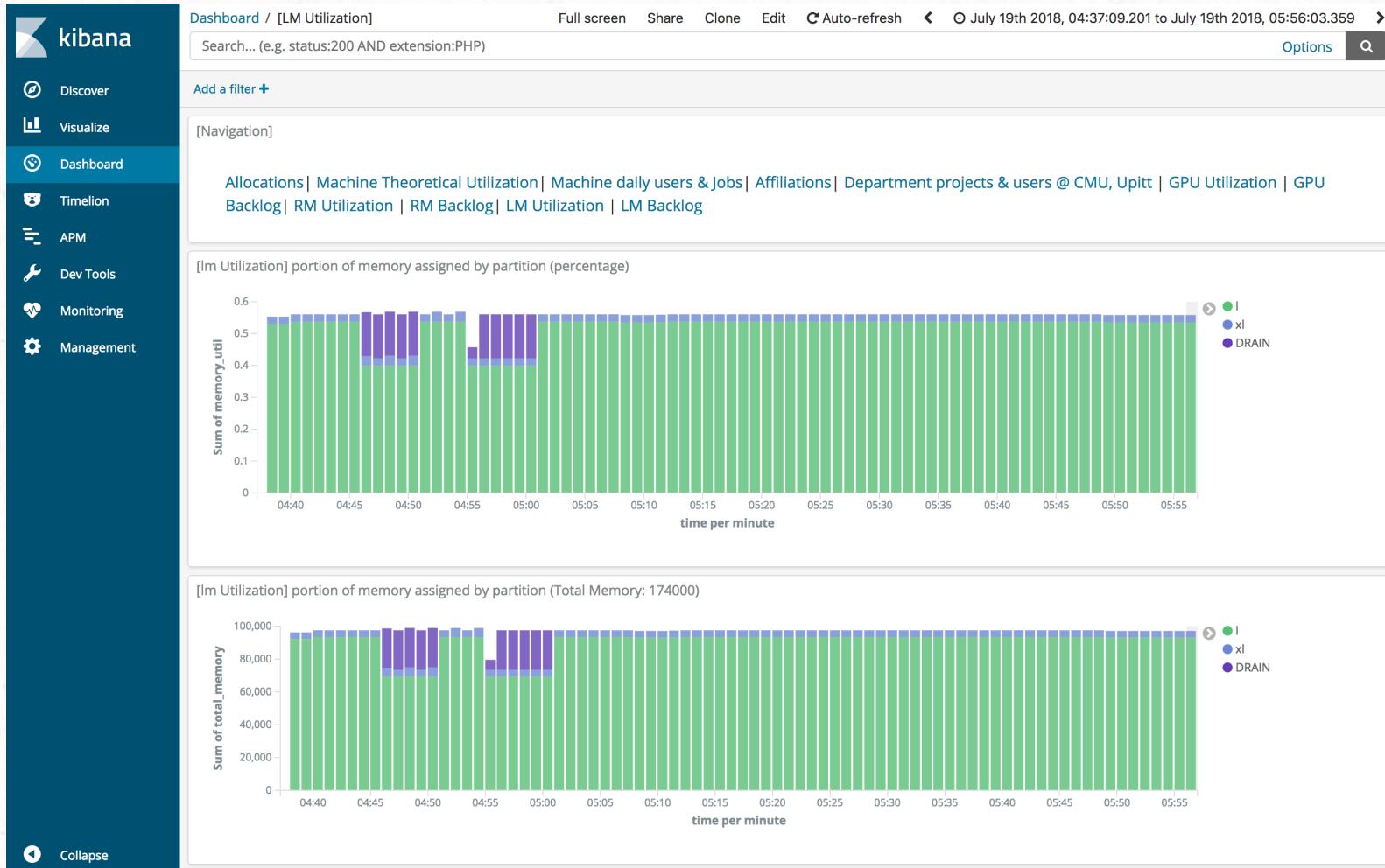
Logstash

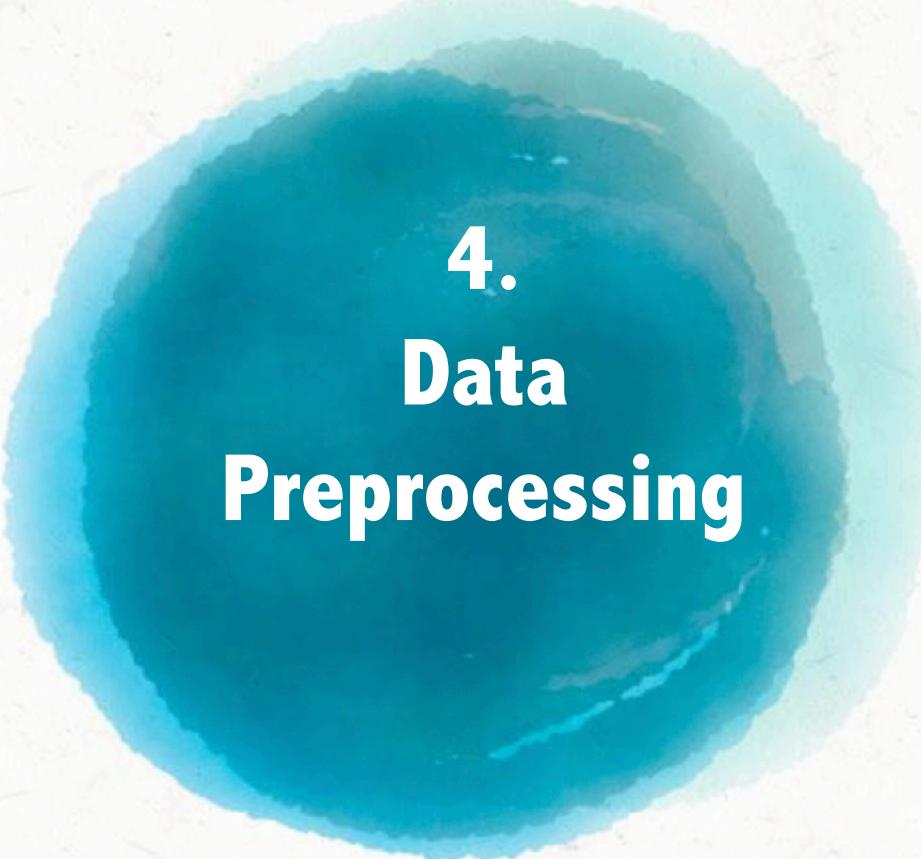


Elasticsearch under the hood • Lucene



Kibana





4. Data Preprocessing

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
AllocTRES
JobName
JobID
User
Partition
Start
State
Submit
ReqMem
Timelimit
NodeList
End
Name: 253, dtype: object

tr561fp
cpu=84,mem=369600M,node=3,billing/gpu=84
CP2K_example
3597006
lizhao
RM
2018-07-18T12:45:12
FAILED
2018-07-18T10:14:15
123200Mn
2-00:00:00
r[107,641,644]
2018-07-20T00:44:09

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

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

Account
JobName
JobID
User
Partition
Start
State
Submit
ReqMem
NodeList
End
Alloc_CPU
Alloc_MEM
Alloc_NODE
nodeArray
ReqTime
Name: 150, dtype: object

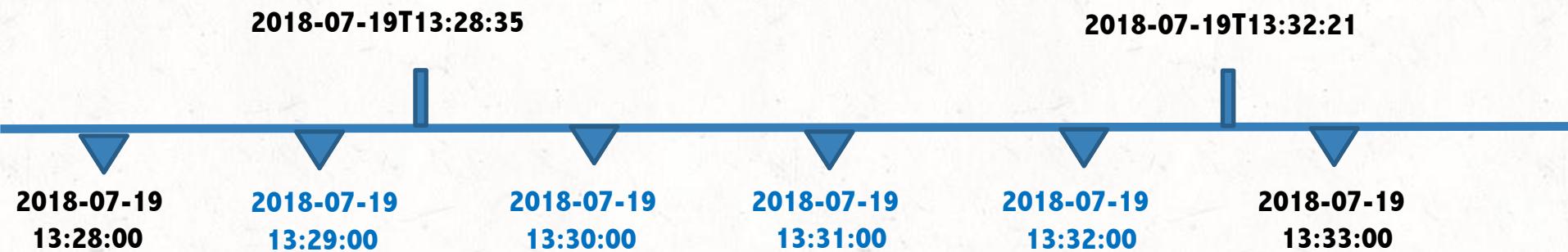
tr561fp
CP2K_example
3597006
lizhao
RM
2018-07-18T12:45:12
FAILED
2018-07-18T10:14:15
123200Mn
r[107,641,644]
2018-07-20T00:44:09
84
369600
3
[107, 641, 644]
172800

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 waits for 10 days to be executed would have $60 \times 24 \times 10 = 14400$ 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

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)`

Original

(8306, 18)

	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		TIMEOUT		1
1	2018-07-19T12:00:00	3000.0		TIMEOUT		1
2	2018-07-19T12:01:00	3000.0		TIMEOUT		1

Backlog

lm_backlog_graph_data.shape (41991, 7)

Original

(4170, 18)

	time	nodeType	mixNode	user	mem	reqTime	jobid
0	2018-07-19T10:56:22	I041	abyss.script	rsleith	3000	1206000.0	1
1	2018-07-19T10:57:22	I041	abyss.script	rsleith	3000	1206000.0	1
2	2018-07-19T10:58:22	I041	abyss.script	rsleith	3000	1206000.0	1

Interactive

lm_interact_graph_data.shape (17, 6)

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

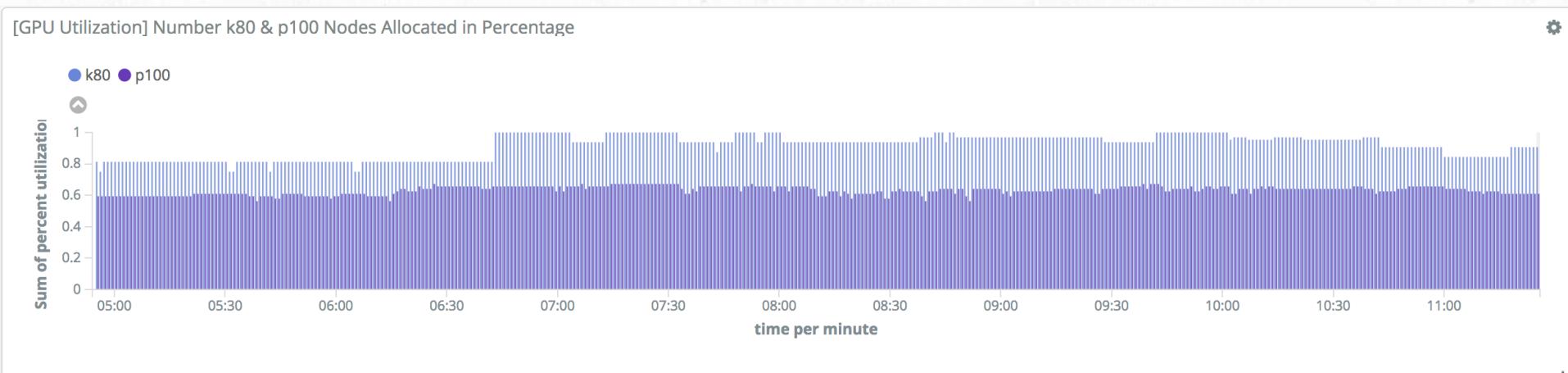


5.

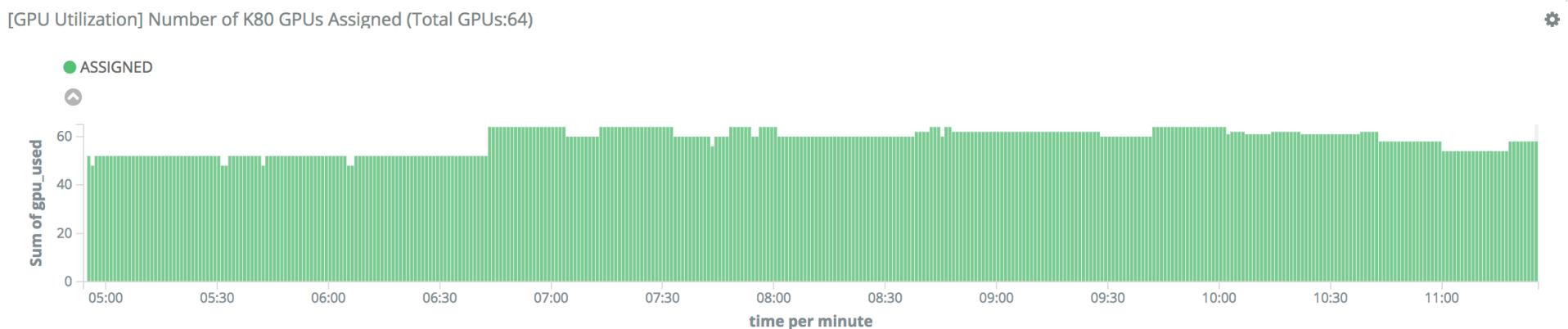
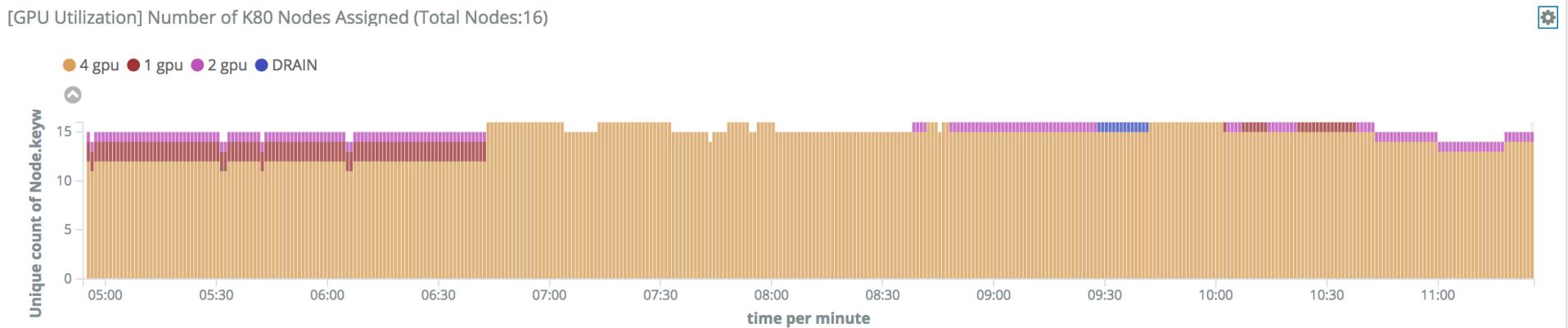
Dashboard

Demo

GPU Utilization



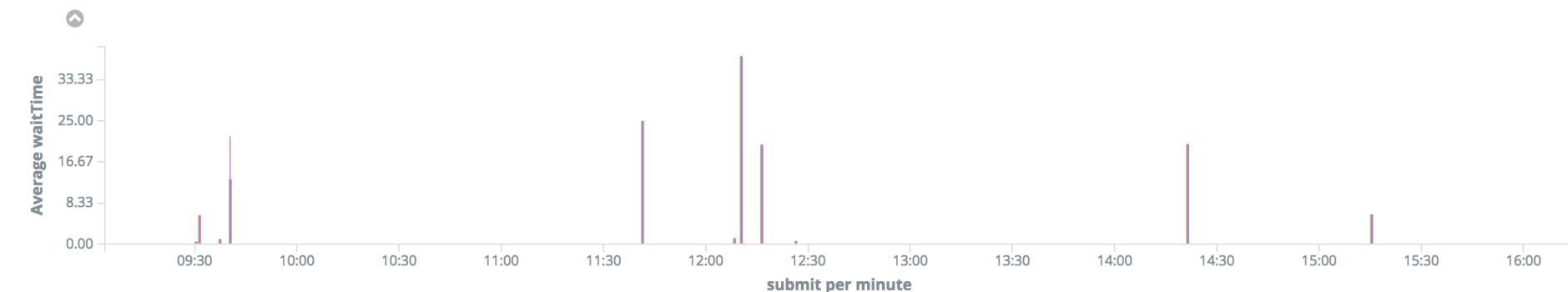
GPU Utilization



GPU Backlog

[GPU Backlog] Average Wait Time

- Average Wait Time (mins)
- Max Wait Time (mins)
- 50th percentile of Median Wait Time (mins)



[GPU Backlog] Number of GPU Interactive Jobs

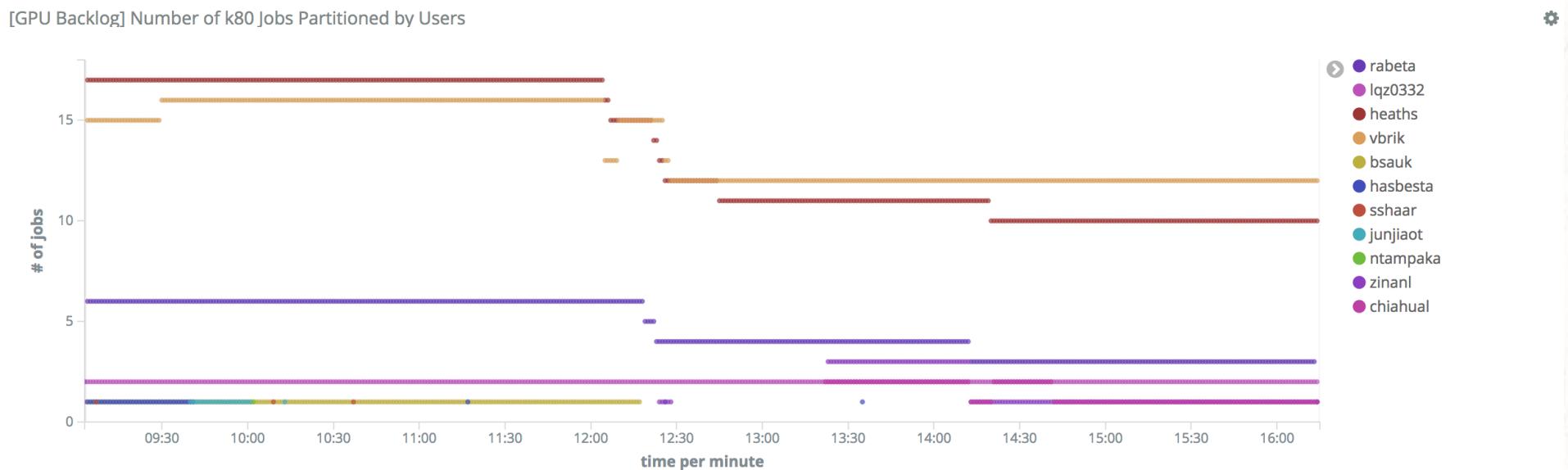
- k80
- p100



GPU Backlog



GPU Backlog





6.

Project Benefits for PSC

“

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

Thanks!



any
questions
?

You can find me at
chiahuai@andrew.cmu.edu