EE3801 Data Engineering Principles

Review

Optimizing Parallel Processing

Lab 8 Techniques

Lab 7

```
(env1) [ec2-user@ip-10-0-5-43 20181105]$ sbatch --
dependency=afterok:11:12:13:14:15:16 /data/src/PyHipp/
consol_jobs.sh
```

Lab 8

```
(env1) [ec2-user@ip-10-0-5-43 picasso]$ bash /data/src/PyHipp/
consol_fsjobs.sh
```

Data Processing on AWS

Cluster Deletion

consol_jobs.sh

```
#!/bin/sh
temp1=($(squeue))
cmd1="sbatch --dependency=afterok:"
counter1=0
for i in "${temp1[@]}"; do
       if [[ "$i" == "queue1" ]]; then
                      id1=${temp1[$counter1-1]}
                      cmd1="${cmd1}${id1}:"
       fi
       counter1=$((counter1+1))
done
cmd1=${cmd1::-1}
cmd1="${cmd1} /data/src/PyHipp/ec2snapshot.sh"

echo $($cmd1)
eval $cmd1
```

```
(env1) [ec2-user@ip-10-0-9-92 data]$ for i in "${temp1[@]}"; do echo $i; done
JOBID
PARTITION
NAME
USER
ST
TIME
NODES
NODELIST(REASON)
233
queue1
rplpl
ec2-user
CF
0:02
1
compute-dy-r5large-1
234
queue1
rse
ec2-user
CF
0:02
1
compute-dy-r5large-1
(env1) [ec2-user@ip-10-0-9-92 data]$
```

cmd1=“sbatch —dependency=afterok:

cmd1=“sbatch —dependency=afterok:233:

cmd1=“sbatch —dependency=afterok:233:234:

cmd1=“sbatch —dependency=afterok:233:234

cmd1=“sbatch —dependency=afterok:233:234 /data/src/PyHipp/ec2snapshot.sh

Optimizing Parallel Processing

Lab 8 Techniques

```
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
    7    queue1    rplpl ec2-user  R       4:31      1 queue1-dy-r52xlarge-1
    8    queue1      rs1 ec2-user  R       4:31      1 queue1-dy-r52xlarge-1
    9    queue1      rs2 ec2-user  R       1:25      1 queue1-dy-r52xlarge-2
   10    queue1      rs3 ec2-user  R       1:25      1 queue1-dy-r52xlarge-3
   11    queue1      rs4 ec2-user  R       1:25      1 queue1-dy-r52xlarge-4
```

(env1) [ec2-user@ip-10-0-5-43 20181105]$ sbatch /data/src/PyHipp/consol_jobs.sh

Optimizing Parallel Processing

Fine-Grained Parallel Processing (226 jobs)

Just-in-time job submission

DPT.objects.processDirs(dirs=None, objtype=pyh.RPLSplit, channel=[*range(1,33)], SkipHPC=False, HPCScriptsDir = '/data/src/PyHipp/', SkipLFP=False, SkipHighPass=False, SkipSort=False);

rplsplit.py — Free Mode

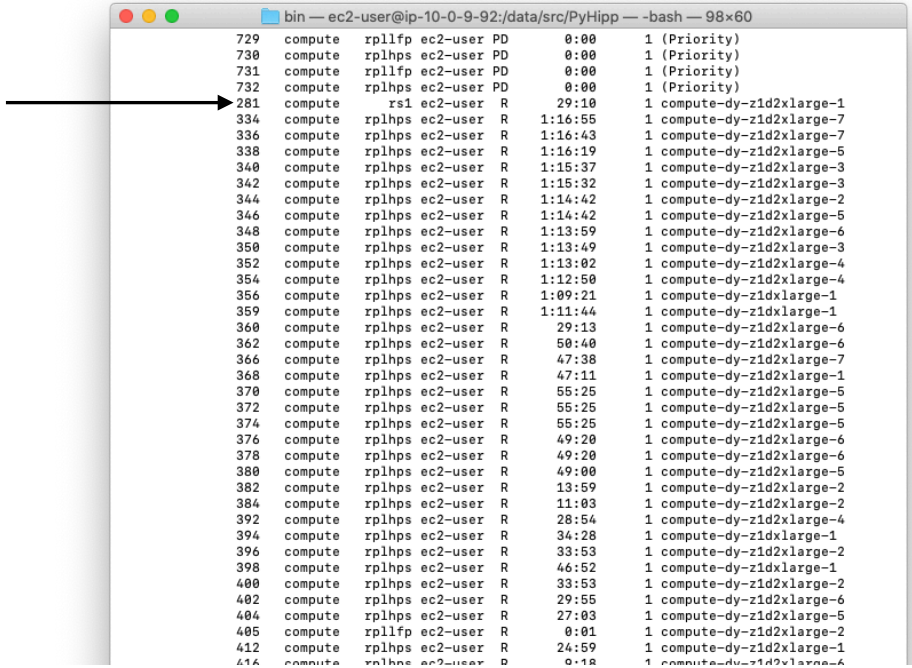
~/Dropbox/Work/Python/PyHipp/PyHipp/rplsplit.py — main — (no function selected)

```
82              os.chdir(channelDir)
83              print('Calling RPLRaw for channel {:03d}'.format(channelNumber))
84              rplraw.RPLRaw(analogData = data, analogInfo = analogInfo, saveLevel = 1)
85              if self.args['SkipHPC']:
86                  if not self.args['SkipLFP']:
87                      print('Calling RPLLFP for channel {:03d}'.format(channelNumber))
88                      rpllfp.RPLLFP(saveLevel = 1)
89                  if not self.args['SkipHighPass']:
90                      print('Calling RPLHighPass for channel {:03d}'.format(channelNumber))
91                      rplhighpass.RPLHighPass(saveLevel = 1)
92                  if DPT.levels.get_level_name('session', os.getcwd()) != 'sessioneye':
93                      if not self.args['SkipSort']:
94                          print('Calling Mountain Sort for channel {:03d}'.format(channelNumber))
95                          # mountain_batch()
96                          # export_mountain_cells()
97              else:
98                  if 'HPCScriptsDir' not in kwargs.keys():
99                      kwargs['HPCScriptsDir'] = ''
100                 if not self.args['SkipLFP']:
101                     print('Adding RPLLFP slurm script for channel {:03d} to job queue'.format(channelNumber))
102                     os.system('sbatch ' + kwargs['HPCScriptsDir'] + 'rpllfp-slurm.sh')
103                 if not self.args['SkipHighPass']:
104                     if not self.args['SkipSort']:
105                         print('Adding RPLHighPass and Mountain Sort slurm script for channel {:03d} to job queue'.format(channelNumber))
106                         os.system('sbatch '+ kwargs['HPCScriptsDir'] + 'rplhighpass-sort-slurm.sh')
107                     else:
108                         print('Adding RPLHighPass slurm script for channel {:03d} to job queue'.format(channelNumber))
109                         os.system('sbatch ' + kwargs['HPCScriptsDir'] + 'rplhighpass-slurm.sh')
```

L: 153 C: 1 — Python — Unicode (UTF-8) — Unix (LF) — Saved: 11/16/20, 5:13:37 PM — 6,438 / 687 / 153 — 100%

Optimizing Parallel Processing

Fine-Grained Parallel Processing (226 jobs)



(env1) [ec2-user@ip-10-0-5-43 20181105]$ sbatch —dependency=afterok:280:281:282:283:284 /data/src/PyHipp/consol_jobs.sh

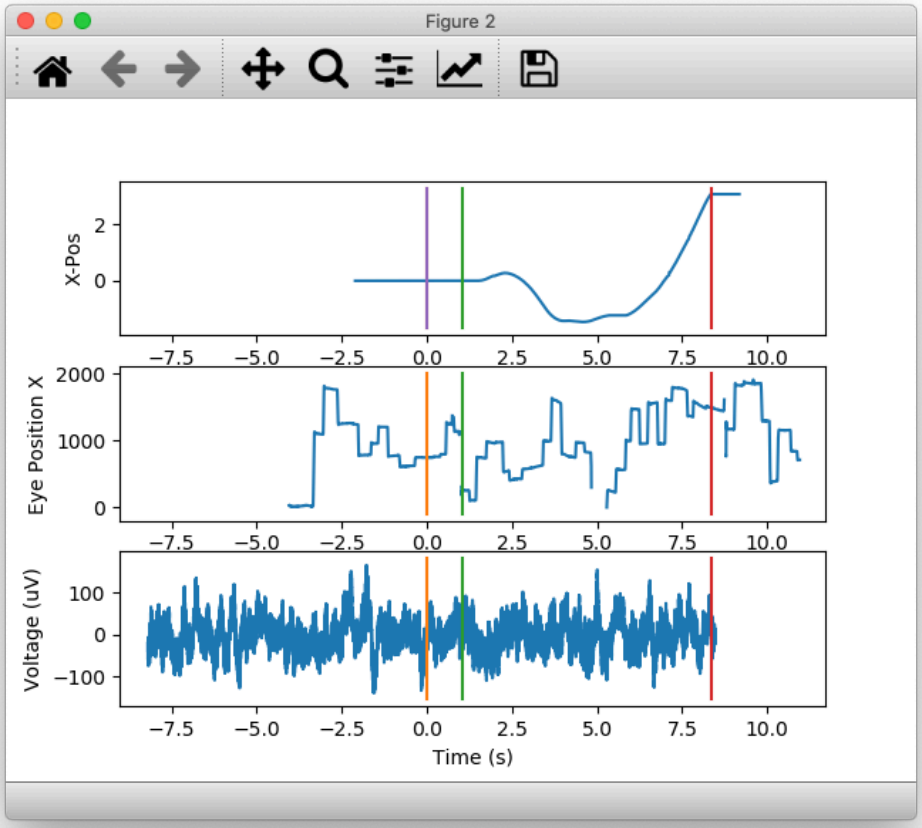
Data Engineering Principles

Review

- Aligning timestamps from different data
- Optimizing data processing pipelines by identifying dependencies
- Automatic head node deletion
- Using DPT to process hierarchically organized data
- Using PanGUI to view data and sets of data easily

Data Acquisition & Processing

Data Timelines



Data Engineering Principles

Review

- Aligning timestamps from different data
- Optimizing data processing pipelines by identifying dependencies
- Automatic head node deletion
- Using DPT to process hierarchically organized data
- Using PanGUI to view data and sets of data easily

Parallel Data Processing

Slurm Script

From Lab 5

```
In[ ]: DPT.objects.processDirs(dirs=None, objtype=pyh.RPLParallel, saveLevel=1)
In[ ]: DPT.objects.processDirs(dirs=None, objtype=pyh.RPLSplit, channel=[9, 31,
34, 56, 72, 93, 119, 120], saveLevel=1)
In[ ]: DPT.objects.processDirs(dirs=None, objtype=pyh.RPLLFP, saveLevel=1)
In[ ]: DPT.objects.processDirs(dirs=None, objtype=pyh.RPLHighPass, saveLevel=1)
In[ ]: DPT.objects.processDirs(dirs=None, objtype=pyh.Unity, saveLevel=1)
In[ ]: pyh.EDFSplit(saveLevel=1)
In[ ]: cd session01
In[ ]: pyh.aligning_objects()
```

```
#!/bin/bash

# Submit this script with: sbatch <this-filename>

#SBATCH --time=1:00:00   # walltime
#SBATCH --ntasks=1   # number of processor cores (i.e. tasks)
#SBATCH --nodes=1   # number of nodes
#SBATCH -J "pipe"   # job name

## /SBATCH -p general # partition (queue)
#SBATCH -o pipe-slurm.%N.%j.out # STDOUT
#SBATCH -e pipe-slurm.%N.%j.err # STDERR

# LOAD MODULES, INSERT CODE, AND RUN YOUR PROGRAMS HERE

python -c "import PyHipp as pyh; import DataProcessingTools as DPT; import os; import time; t0 = time.time(); print(time.localtime());
DPT.objects.processDirs(dirs=None, objtype=pyh.RPLParallel, saveLevel=1); DPT.objects.processDirs(dirs=None, objtype=pyh.RPLSplit,
channel=[9, 31, 34, 56, 72, 93, 119, 120]); DPT.objects.processDirs(dirs=None, objtype=pyh.RPLLFP, saveLevel=1);
DPT.objects.processDirs(dirs=None, objtype=pyh.RPLHighPass, saveLevel=1); DPT.objects.processDirs(dirs=None, objtype=pyh.Unity,
saveLevel=1); pyh.EDFSplit(); os.chdir('session01'); pyh.aligning_objects(); pyh.raycast(1); DPT.objects.processDirs(level='channel',
cmd='import PyHipp as pyh; from PyHipp import mountain_batch; mountain_batch.mountain_batch(); from PyHipp import
export_mountain_cells; export_mountain_cells.export_mountain_cells();'); print(time.localtime()); print(time.time()-t0);"
```

Called from /data/picasso/20181105

Parallel Data Processing

Parallel Processing

	RPLParallel	RPLSplit	RPLLFP	RPLHighPass	Unity	EDFSplit
RPLParallel						
RPLSplit						
RPLLFP		√				
RPLHighPass		√				
Unity	√					
EDFSplit	√					
aligning_objects	√				√	√
raycast					√	√
mountain_batch				√		

RPLParallel	RPLSplit
Unity	RPLLFP
EDFSplit	RPLHighPass
aligning_objects	mountain_batch
raycast	

Optimizing Parallel Processing

Coarse-Grain Parallel Processing (5 jobs)

Time

RPLParallel	RPLSplit		RPLSplit		RPLSplit		RPLSplit	
Unity	session01	channel001	session01	channel033	session01	channel065	session01	channel097
EDFSplit	sessioneye	channel002	sessioneye	channel034	sessioneye	channel066	sessioneye	channel098
aligning_objects		…		…		…		…
raycast		channel032		channel064		channel096		channel124
	RPLLFP		RPLLFP		RPLLFP		RPLLFP	
	session01	channel001	session01	channel033	session01	channel065	session01	channel097
	sessioneye	channel002	sessioneye	channel034	sessioneye	channel066	sessioneye	channel098
		…		…		…		…
		channel032		channel064		channel096		channel124
	RPLHighPass		RPLHighPass		RPLHighPass		RPLHighPass	
	session01	channel001	session01	channel033	session01	channel065	session01	channel097
	sessioneye	channel002	sessioneye	channel034	sessioneye	channel066	sessioneye	channel098
		…		…		…		…
		channel032		channel064		channel096		channel124
	mountain_batch		mountain_batch		mountain_batch		mountain_batch	
	session01	channel001	session01	channel033	session01	channel065	session01	channel097
		channel002		channel034		channel066		channel098
		…		…		…		…
		channel032		channel064		channel096		channel124

Optimizing Parallel Processing

Fine-Grain Parallel Processing

Time

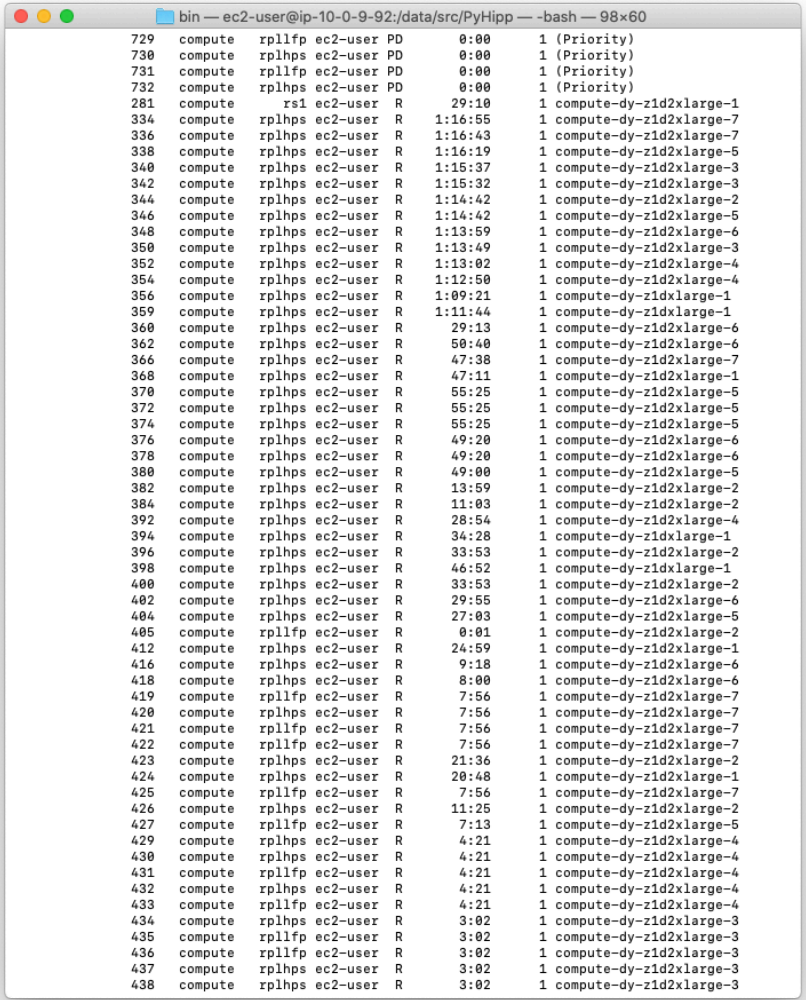
RPLParallel	RPLSplit	
Unity	sessioneye	channel001
EDFSplit		channel002
aligning_objects		…
raycast		channel124
	RPLLFP	
	sessioneye	channel001
		channel002
		…
		channel124
	RPLHighPass	
	sessioneye	channel001
		channel002
		…
		channel124

RPLSplit		RPLLFP	RPLHPS	RPLSplit		RPLLFP	RPLHPS	RPLSplit		RPLLFP	RPLHPS	RPLSplit		RPLLFP	RPLHPS
session01	channel001			session01	channel033			session01	channel065			session01	channel097		
	…				…				…				…		
	channel032				channel064				channel096				channel124		

Jobs: 6 + 110 + 110 = 226

Optimizing Parallel Processing

Fine-Grain Parallel Processing (226 jobs)



Cannot actually run 64 jobs as the Master Node uses 2 CPUs and the EC2 instance also uses 2 CPUs

Data Engineering Principles

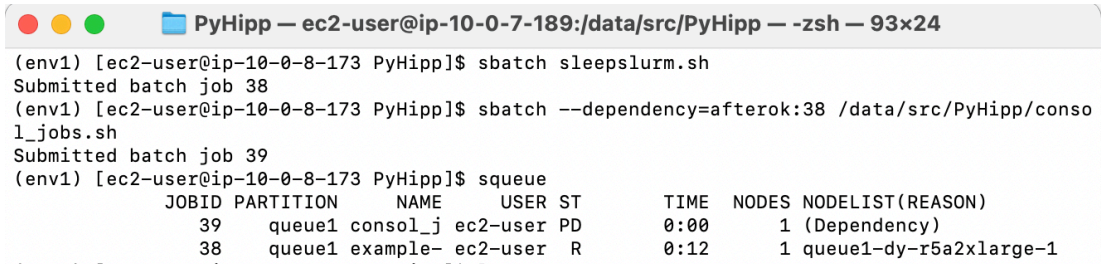
Review

- Aligning timestamps from different data
- Optimizing data processing pipelines by identifying dependencies
- Automatic head node deletion
- Using DPT to process hierarchically organized data
- Using PanGUI to view data and sets of data easily

Data Processing on AWS

Cluster Deletion

- Use slurm job dependencies to initiate snapshot:

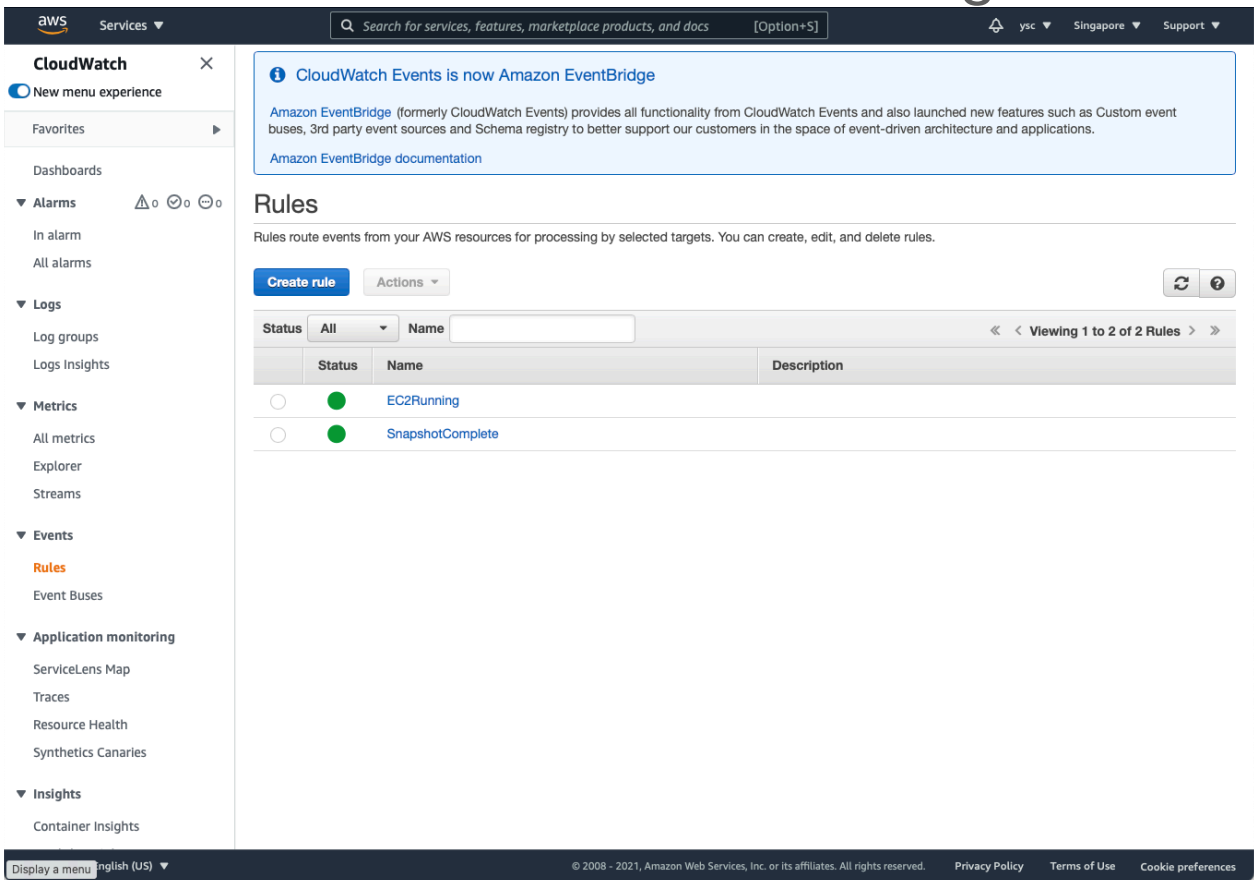


  - ssh -o StrictHostKeyChecking=no -i /data/MyKeyPair.pem ec2-user@xx.xx.xx.xx “source ~/.bash_profile; pcluster update-compute-fleet --status STOP_REQUESTED -n MyCluster01; ~/update_snapshot.sh data 2 MyCluster01”
  - CloudWatch watch for snapshot completion
  - Lambda function used to delete head node
  - Still have to call pcluster delete-cluster to remove from list

Data Processing on AWS

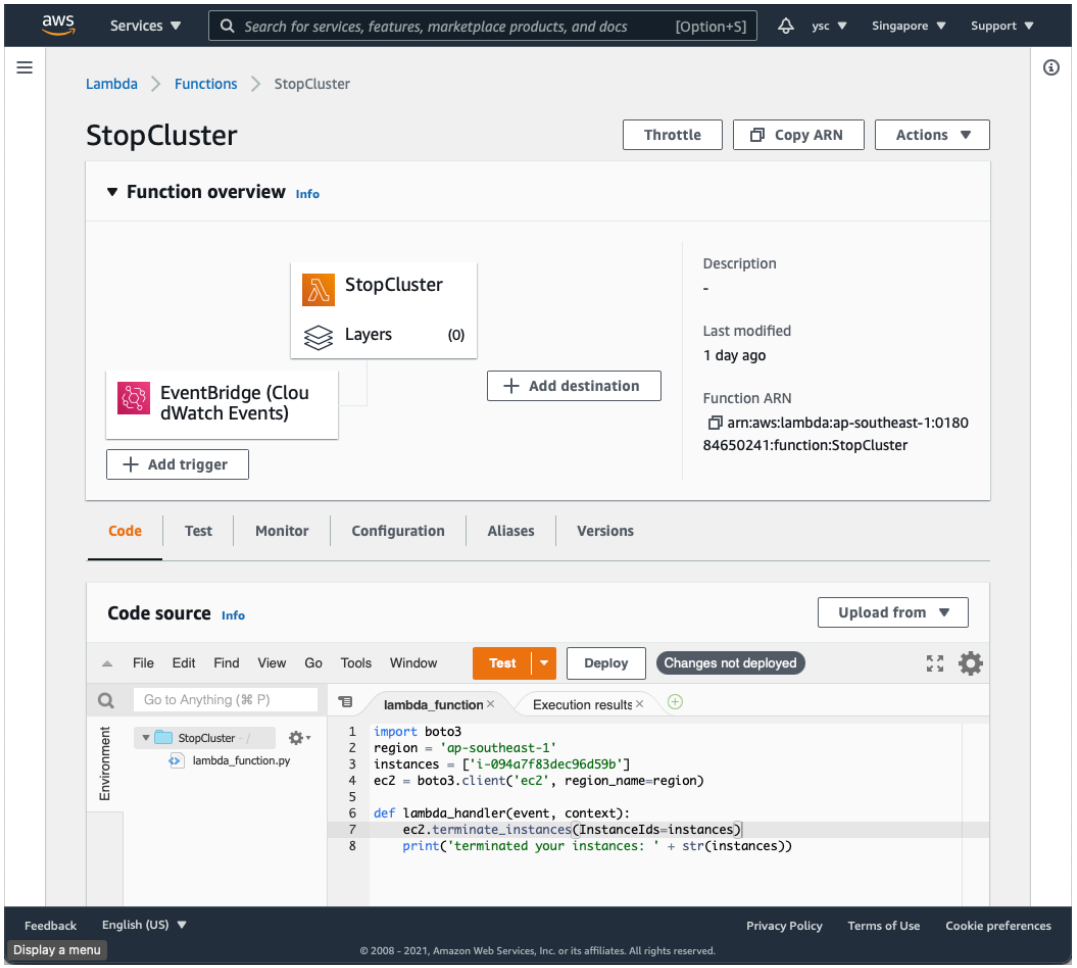
Cluster Deletion

Can we use CloudWatch/EventBridge instead?



Data Processing on AWS

Cluster Deletion



Data Engineering Principles

Review

- Aligning timestamps from different data
- Optimizing data processing pipelines by identifying dependencies
- Automatic head node deletion
- Using DPT to process hierarchically organized data
- Using PanGUI to view data and sets of data easily

Parallel Data Processing

Data Processing Tools Levels

DataProcessingTools/DataProcessingTools/config.json

```
{"subjects": {"pattern": "([a-zA-Z]+)","order":0},
      /data
   "subject": {"pattern": "([a-zA-Z]+)", "order": 1},
      picasso
   "day": {"pattern": "([0-9]+)", "order": 2},
      20181105
   "session": {"pattern": "(session)([a-z0-9]+)","order": 3},
      session01
   "array": {"pattern": "(array)([0-9]+)", "order": 4},
      array01
   "channel": {"pattern": "(channel)([0-9]+)", "order": 5},
      channel009
   "cell": {"pattern": "(cell)([0-9]+)", "order": 6}
      cell01
}
```

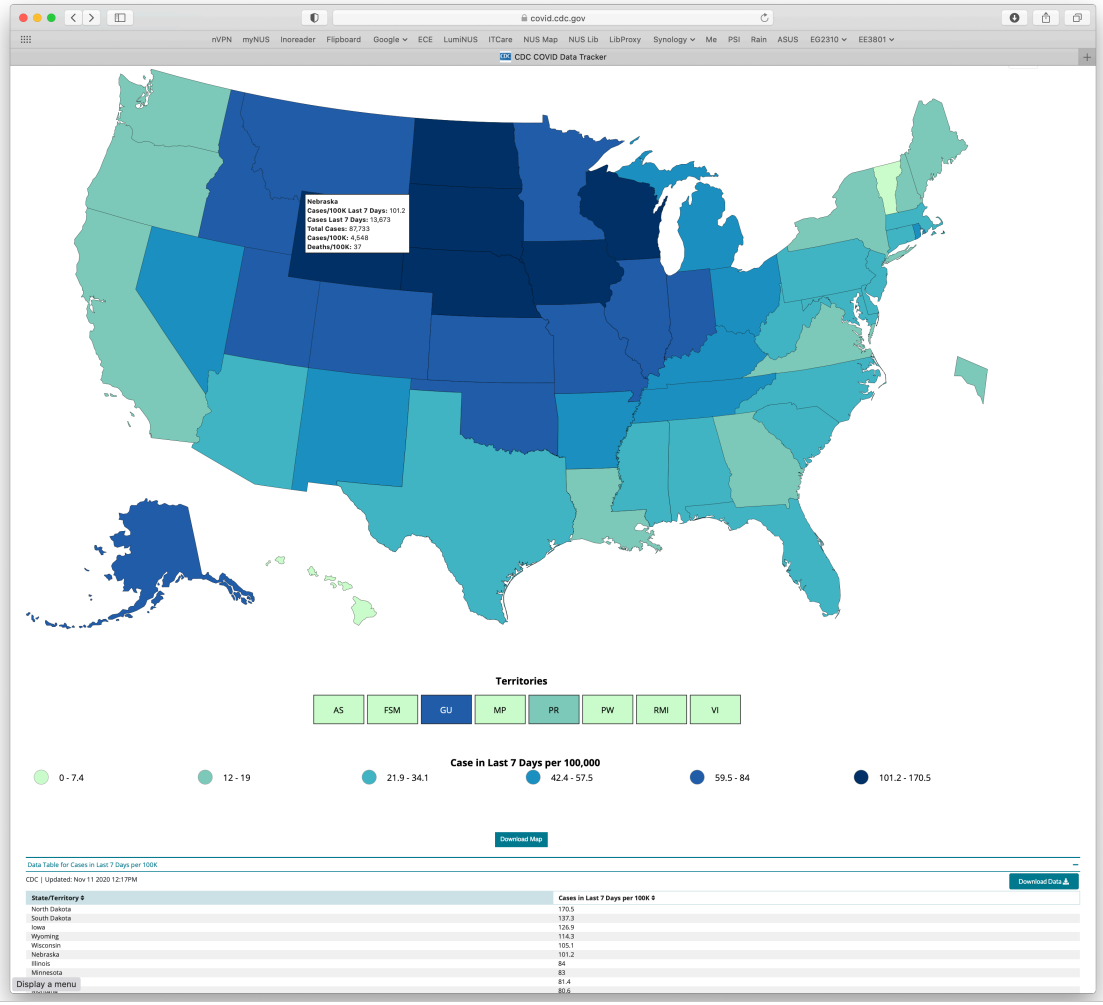
Data Acquisition & Processing

20181105				
	…			
	session01			
		…		
		array01		
			channel009	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel031	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array02		
			channel034	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel056	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array03		
			channel072	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel093	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array04		
			channel119	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel120	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl

20181105				
	…			
	sessioneye			
		…		
		array01		
			channel009	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel031	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array02		
			channel034	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel056	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array03		
			channel072	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel093	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
		array04		
			channel119	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl
			channel120	
				rplraw_xxxx.hkl
				rpllfp_xxxx.hkl

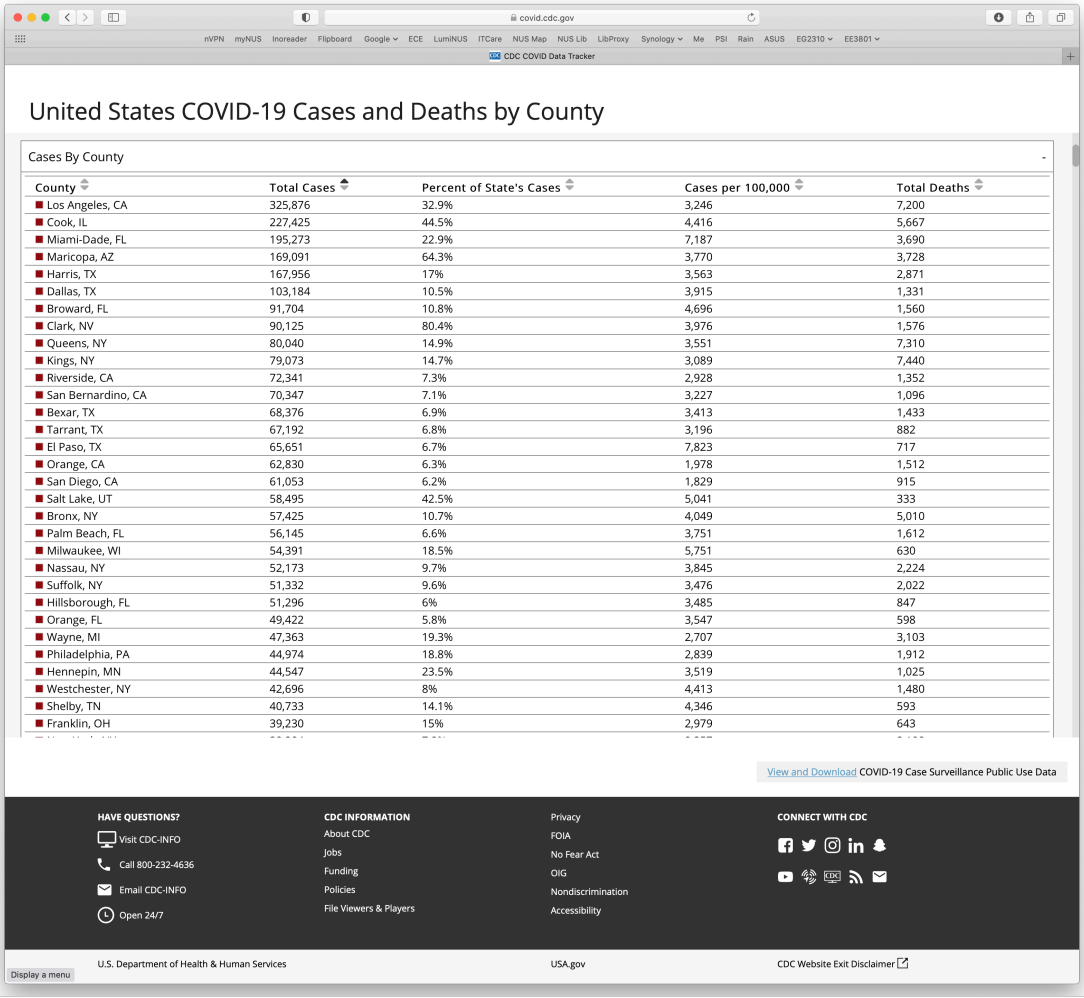
Data Acquisition & Processing

Hierarchically Organized Data



Data Acquisition & Processing

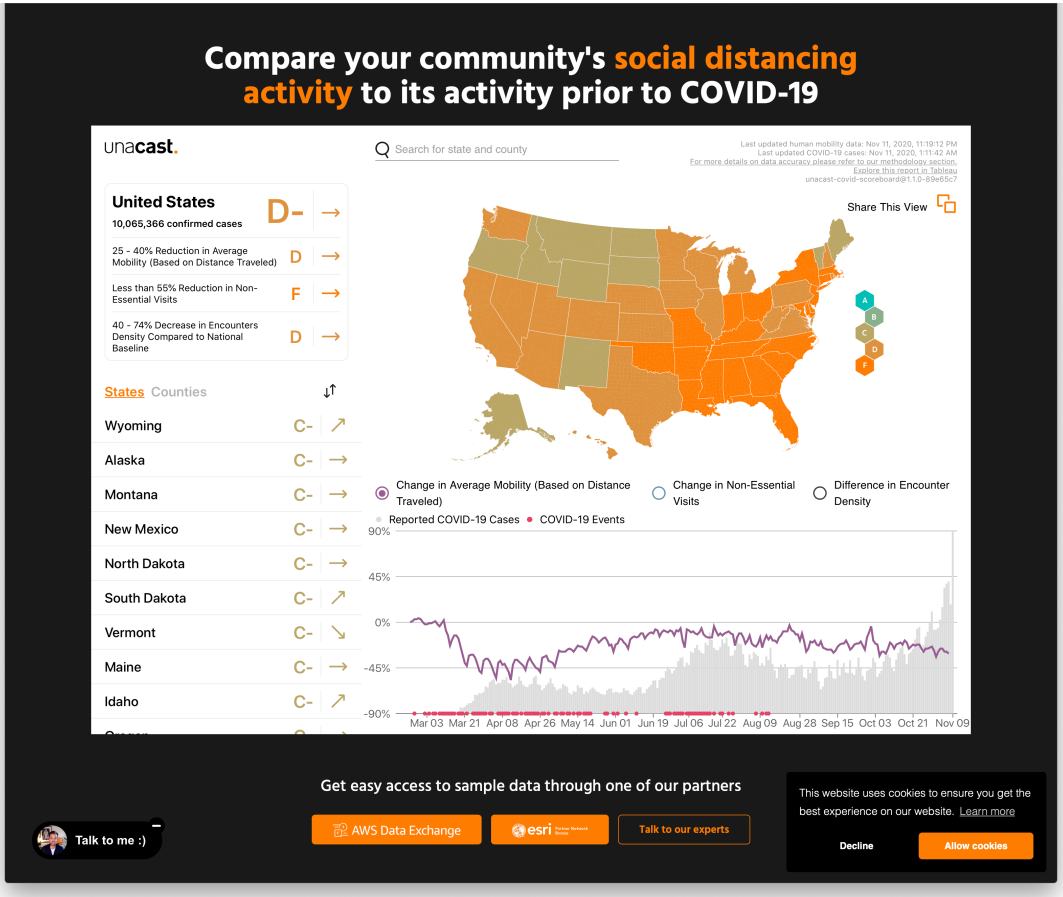
Hierarchically Organized Data



County	Total Cases	Percent of State's Cases	Cases per 100,000	Total Deaths
Los Angeles, CA	325,876	32.9%	3,246	7,200
Cook, IL	227,425	44.5%	4,416	5,667
Miami-Dade, FL	195,273	22.9%	7,187	3,690
Maricopa, AZ	169,091	64.3%	3,770	3,728
Harris, TX	167,956	17%	3,563	2,871
Dallas, TX	103,184	10.5%	3,915	1,331
Broward, FL	91,704	10.8%	4,696	1,560
Clark, NV	90,125	80.4%	3,976	1,576
Queens, NY	80,040	14.9%	3,551	7,310
Kings, NY	79,073	14.7%	3,089	7,440
Riverside, CA	72,341	7.3%	2,928	1,352
San Bernardino, CA	70,347	7.1%	3,227	1,096
Bexar, TX	68,376	6.9%	3,413	1,433
Tarrant, TX	67,192	6.8%	3,196	882
El Paso, TX	65,651	6.7%	7,823	717
Orange, CA	62,830	6.3%	1,978	1,512
San Diego, CA	61,053	6.2%	1,829	915
Salt Lake, UT	58,495	42.5%	5,041	333
Bronx, NY	57,425	10.7%	4,049	5,010
Palm Beach, FL	56,145	6.6%	3,751	1,612
Milwaukee, WI	54,391	18.5%	5,751	630
Nassau, NY	52,173	9.7%	3,845	2,224
Suffolk, NY	51,332	9.6%	3,476	2,022
Hillsborough, FL	51,296	6%	3,485	847
Orange, FL	49,422	5.8%	3,547	598
Wayne, MI	47,363	19.3%	2,707	3,103
Philadelphia, PA	44,974	18.8%	2,839	1,912
Hennepin, MN	44,547	23.5%	3,519	1,025
Westchester, NY	42,696	8%	4,413	1,480
Shelby, TN	40,733	14.1%	4,346	593
Franklin, OH	39,230	15%	2,979	643

Data Acquisition & Processing

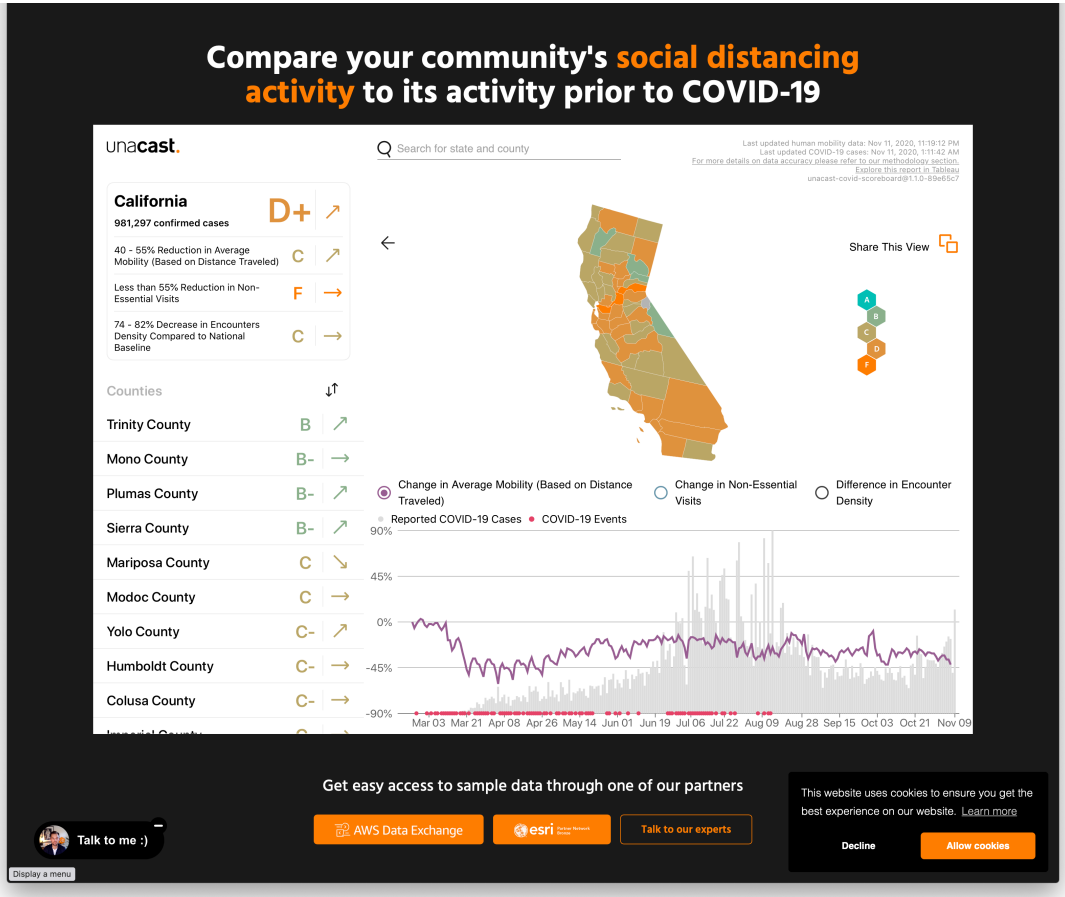
Hierarchically Organized Data



https://www.unacast.com/covid19/social-distancing-scoreboard

Data Acquisition & Processing

Hierarchically Organized Data



https://www.unacast.com/covid19/social-distancing-scoreboard

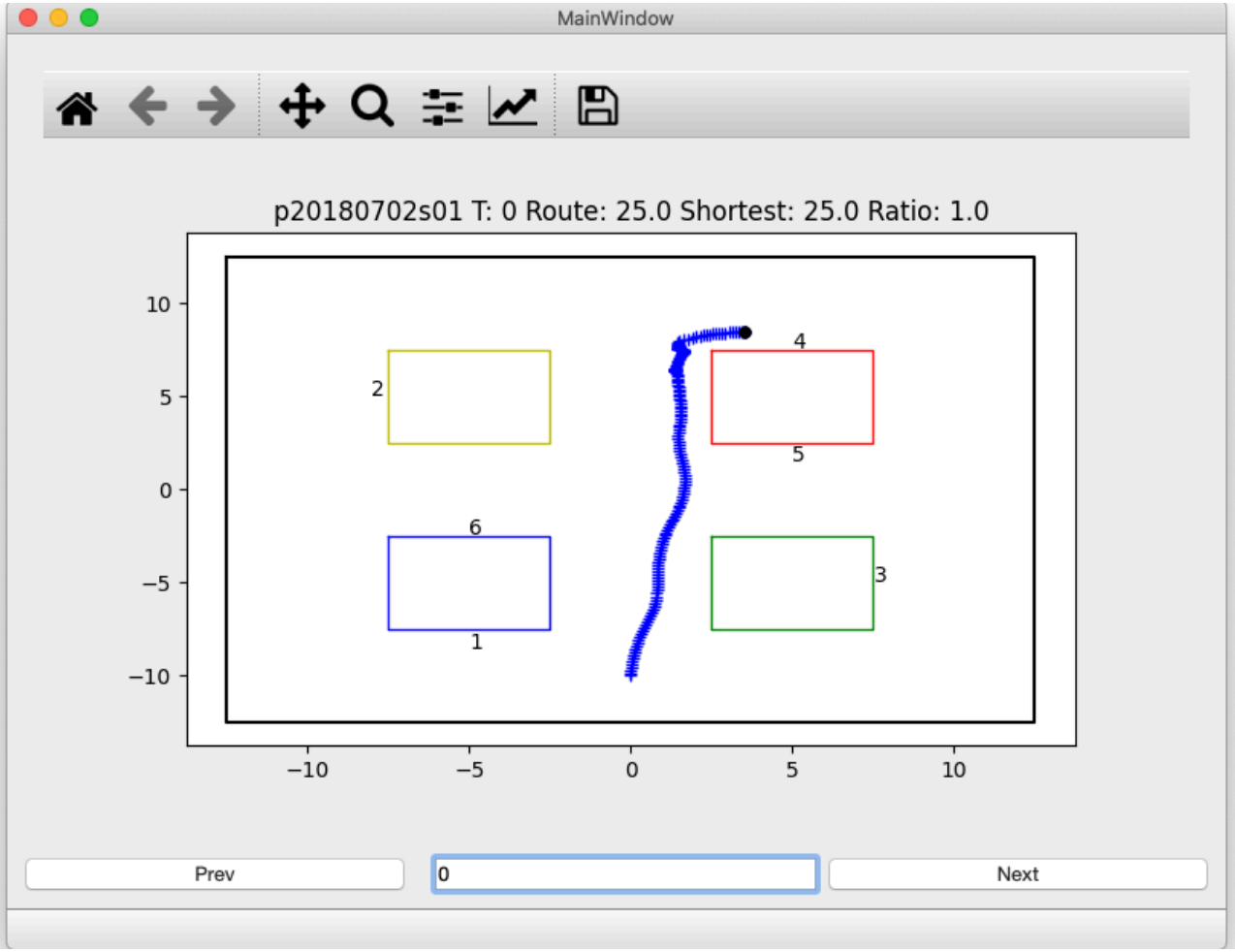
Data Engineering Principles

Review

- Aligning timestamps from different data
- Optimizing data processing pipelines by identifying dependencies
- Using DPT to process hierarchically organized data
- Using PanGUI to view data and sets of data easily

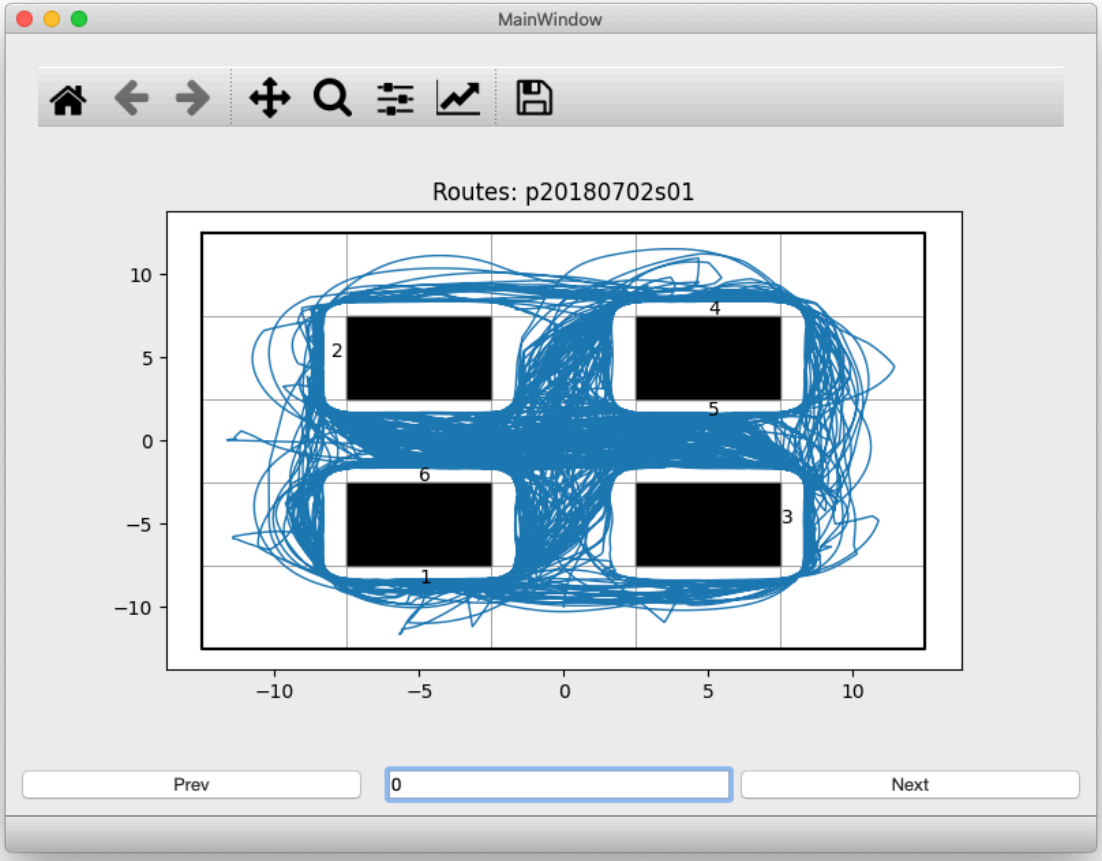
High-Throughput Visualization

PanGUI



High-Throughput Visualization

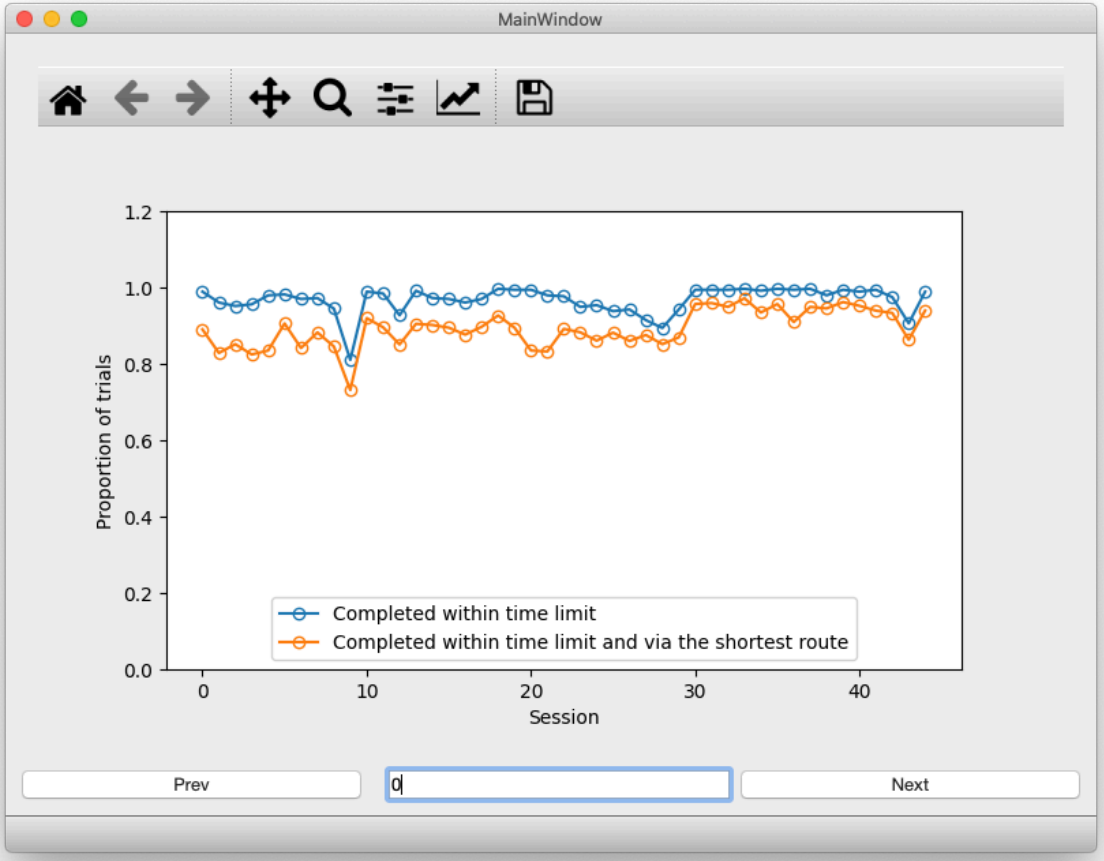
PanGUI



403 Trials

High-Throughput Visualization

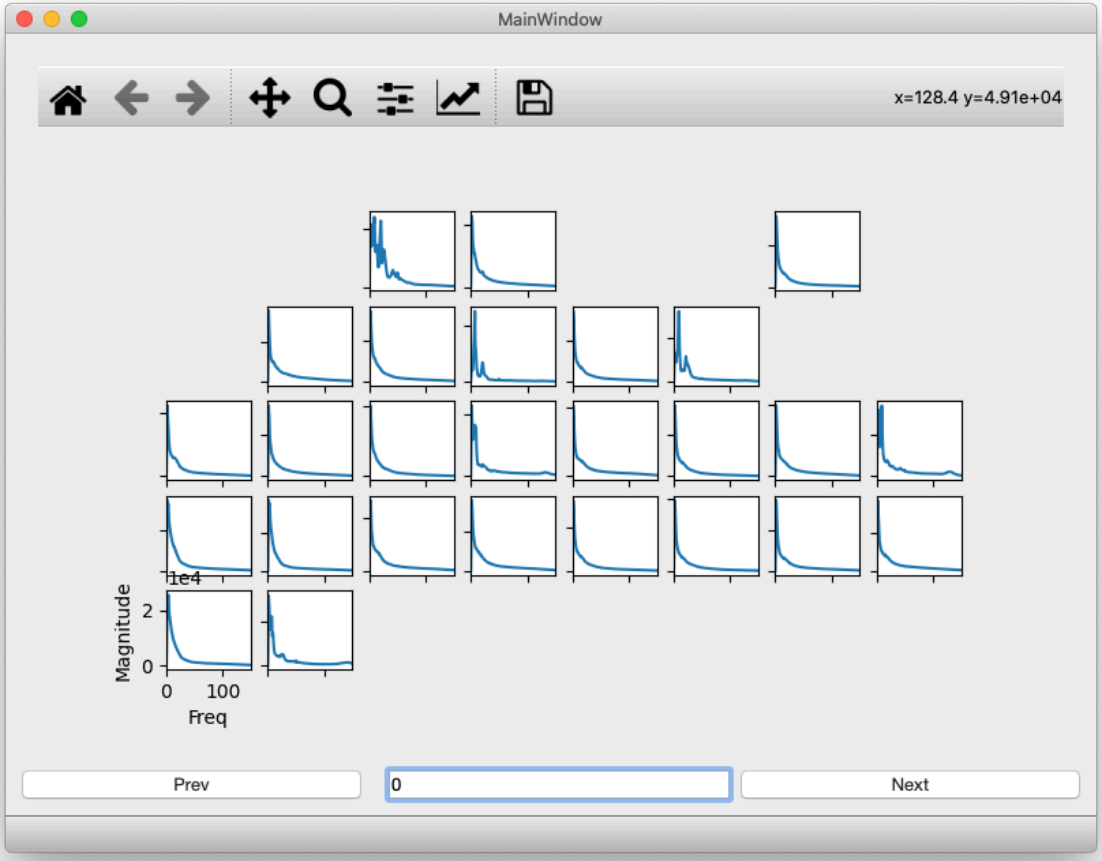
PanGUI



18,305 Trials

High-Throughput Visualization

PanGUI



32 channels

Quiz Preparations

Quiz on Nov 16 from 3 to 4 pm

- Before the quiz
  - Modify cluster-config.yaml on your EC2 instance to use only one compute node (keep the instance type the same as the one you used for the labs)

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: queue1
    ComputeResources:
    - Name: r5-2xlarge
      InstanceType: r5.2xlarge
      MinCount: 1
      MaxCount: 1
```

  - In order to reduce time spent copying files during the quiz, you will be given a Snapshot ID on Canvas Announcements 30 mins before the quiz to copy and subsequently use during cluster creation
  - Create the cluster from your EC2 instance before the quiz (change instance type for the compute node if necessary)
  - Do not perform any processing before the start of the quiz
  - Link to quiz instructions will be sent via Canvas Announcements at 3 pm

Quiz Preparations

- Past year quizzes and solutions have been posted on Canvas
- Scripts from Lab 8 have been added to the upstream repository

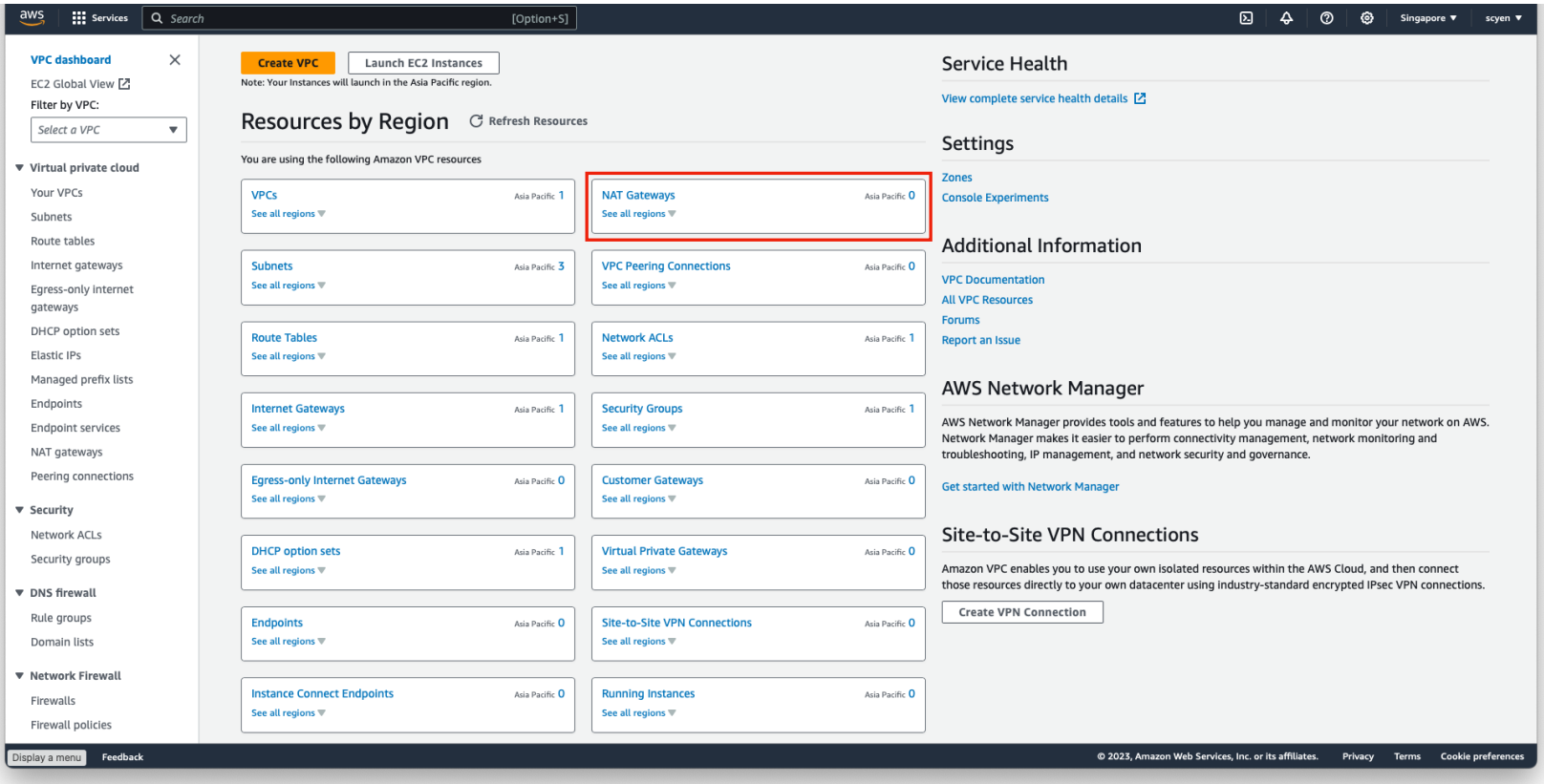
Data Engineering Principles

Wrap Up

- Cleaning up
  - pcluster list
  - Delete clusters by doing: pcluster delete ClusterName
  - Delete EC2 instance using AWS Dashboard
  - Delete EBS snapshots using AWS Dashboard
  - Delete volumes
  - Delete NAT Gateways in VPC

Data Engineering Principles

Wrap Up



Questions?