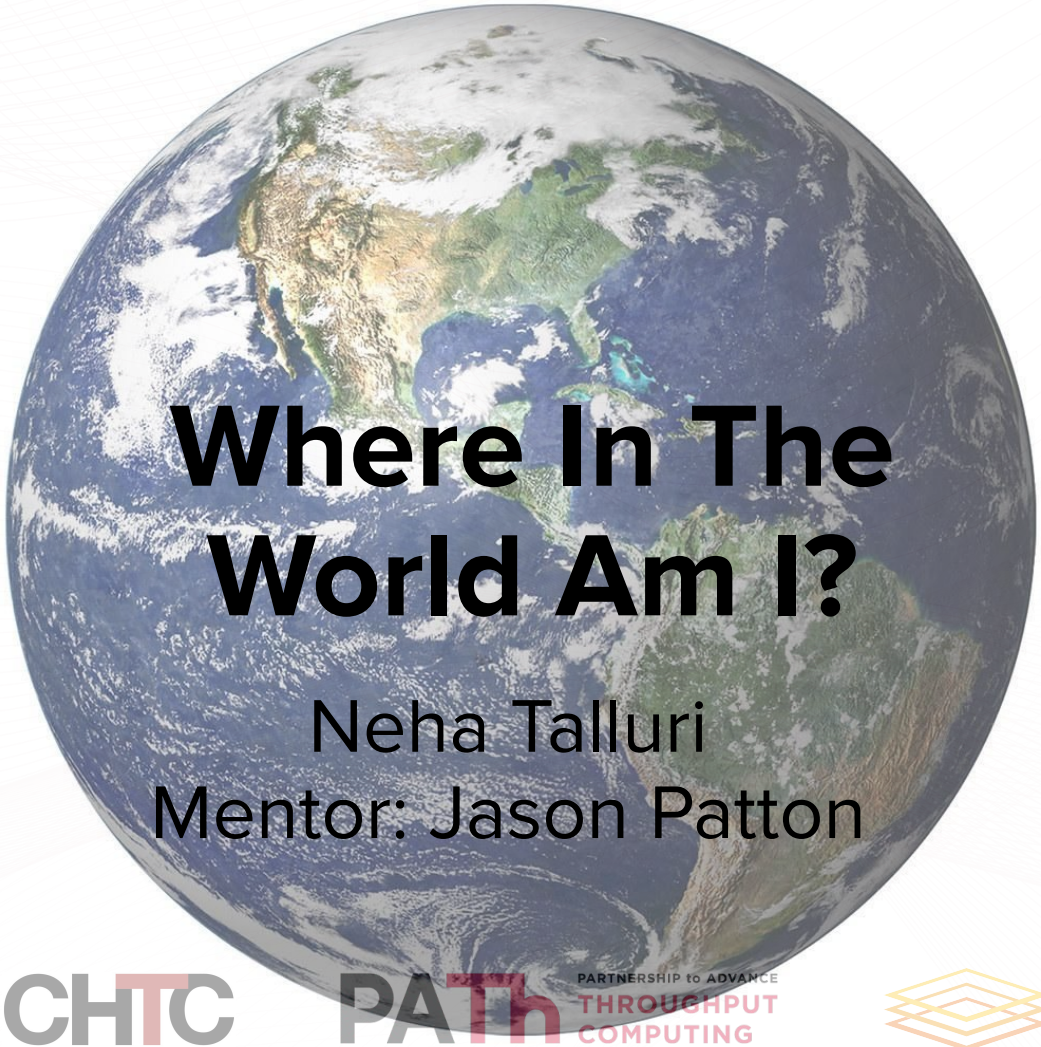
2024 Summer Fellows

CHTC

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING



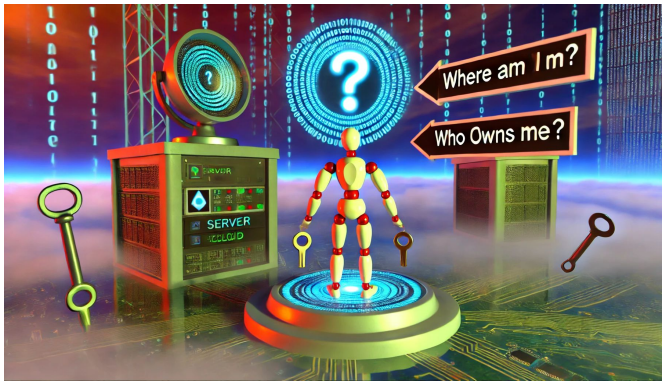
Where In The World Am I?

Neha Talluri
Mentor: Jason Patton

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

What is a Glidein?

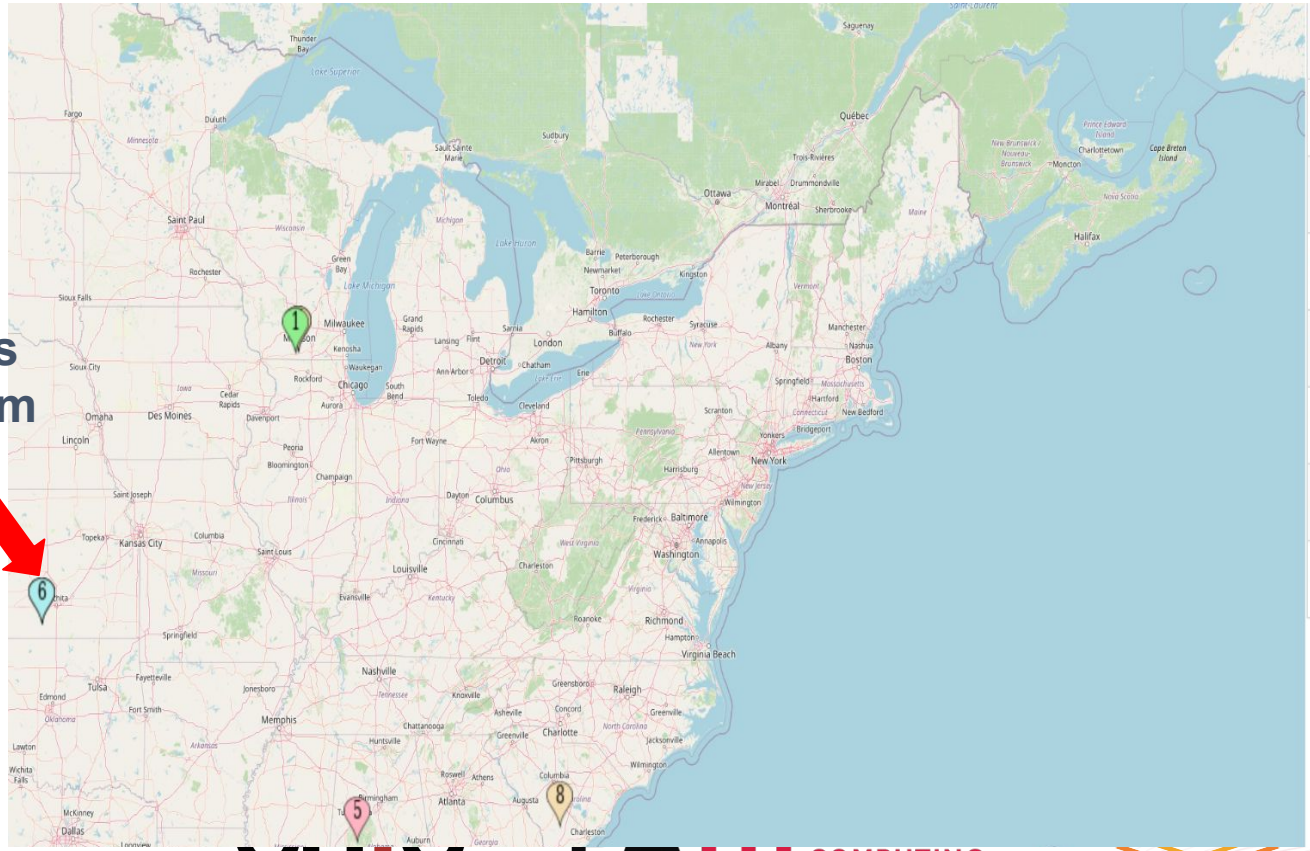
- A glidein runs on shared resources to create more execution points
- Glideins assess the worker node they are on by gathering information and utilizing pre-configured data
- Location of a Glidein is currently not discovered or provided
  - Institutions may not be co-located with the resource



CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

IP Geolocation Problematic

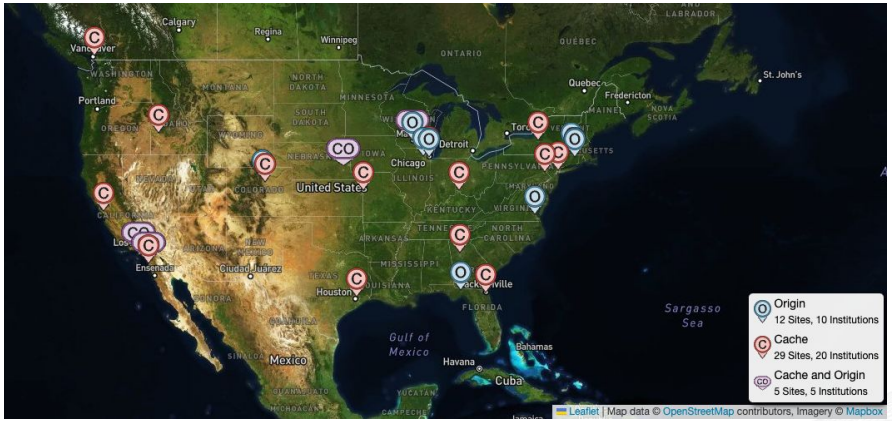
Kansas problem



1	43.0344, -89.5007
2	43.0782, -89.4075
3	37.751, -97.822
4	37.751, -97.822
5	33.1826, -87.4827
6	37.751, -97.822
7	43.0782, -89.4075
8	33.4987, -80.8466

“Where Am I?”

- Location: is the distance to some known entity
- Distance: Network latency and hops
- Answer “Where are glideins in relation to known entities?”
  - Entities = OSDF Caches



CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

What I've Been Doing

- Learning the glideinWMS system through running OSPool jobs and kicking off glideins
- Developing glidein scripts to gather network information and advertise this information to the machine ad and glidein logs
- Figuring out how to answer "Where am I?"
  - Testing IP geolocation using different IP addresses
  - Figuring out how to use tracepath to figure out latency and hops

```
Hop #,Hostname,IP,Latency,Aysmm,Aysmm_Grade
1?,[LOCALHOST],pmtu,1500,,
1,10.5.255.253,10.5.255.253,0.677ms,,
1,10.5.255.253,10.5.255.253,0.730ms,,
2,10.30.29.34,10.30.29.34,0.253ms,,
3,10.30.29.44,10.30.29.44,0.611ms,asymm,2
4,128.230.61.33,128.230.61.33,0.950ms,asymm,3
5,syr-9208-su.nysernet.net,199.109.9.5,0.804ms,asymm,4
6,buf-9208-syr-9208.nysernet.net,199.109.7.194,4.489ms,asymm,5
7,I2-CLEV-buf-9208.nysernet.net,199.109.11.34,9.830ms,asymm,6
8,fourhundredge-0-0-0-2.4079.core2.eqch.net.internet2.edu,163.253.2.17,26.712ms,asymm,11
9,fourhundredge-0-0-0-2.4079.core2.chic.net.internet2.edu,163.253.2.18,26.276ms,asymm,10
10,fourhundredge-0-0-0-1.4079.core1.kans.net.internet2.edu,163.253.1.245,25.641ms,asymm,9
```

Questions?

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

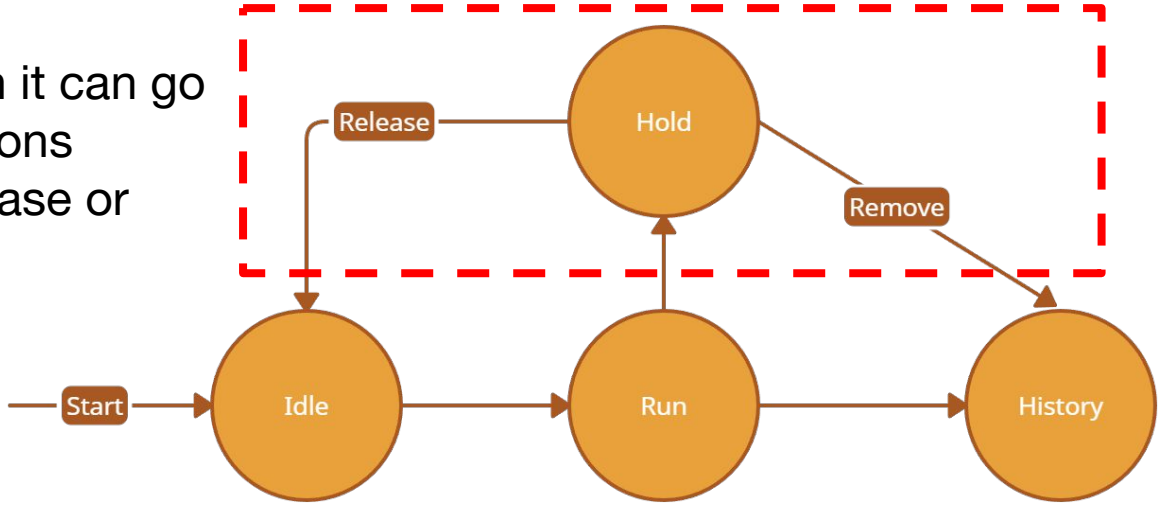
Machine Learning for OSPool Failure Classification

Thinh Nguyen
Mentor: Justin Hiemstra

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

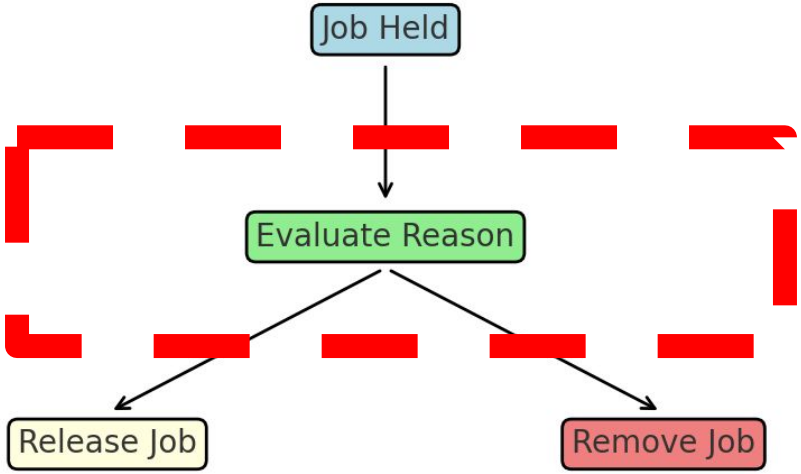
An Example

- Lifecycle of a job
- During a job's execution it can go on hold for various reasons
- User's discretion to release or remove it



Using AI to Make the Inference

- Why AI?
  - OSPool is a Dynamic system
  - Continual learning



Usage

```
[tdnguyen25@ap2002 logs]$ inference 1741531 ./job.log
```

How?

- Each job has a log file describing its lifecycle
- Format these logs into a time-series structure
- Use model that accounts for temporal patterns
  - e.g. Long Short-Term Memory (LSTM) neural network

Bonus

- Can the model provide information as to why the job went on hold?

Thank you,

Questions?

https://github.com/super10099/Machine-Learning-for-OSPool-Failure-Classification

Expanding Pelican Origin Monitoring

Patrick Brophy

CHTC

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Who Am I?

- My name is Patrick Brophy
- I am a senior at UW Madison studying computer science
- CHTC Fellow working with Haoming Meng

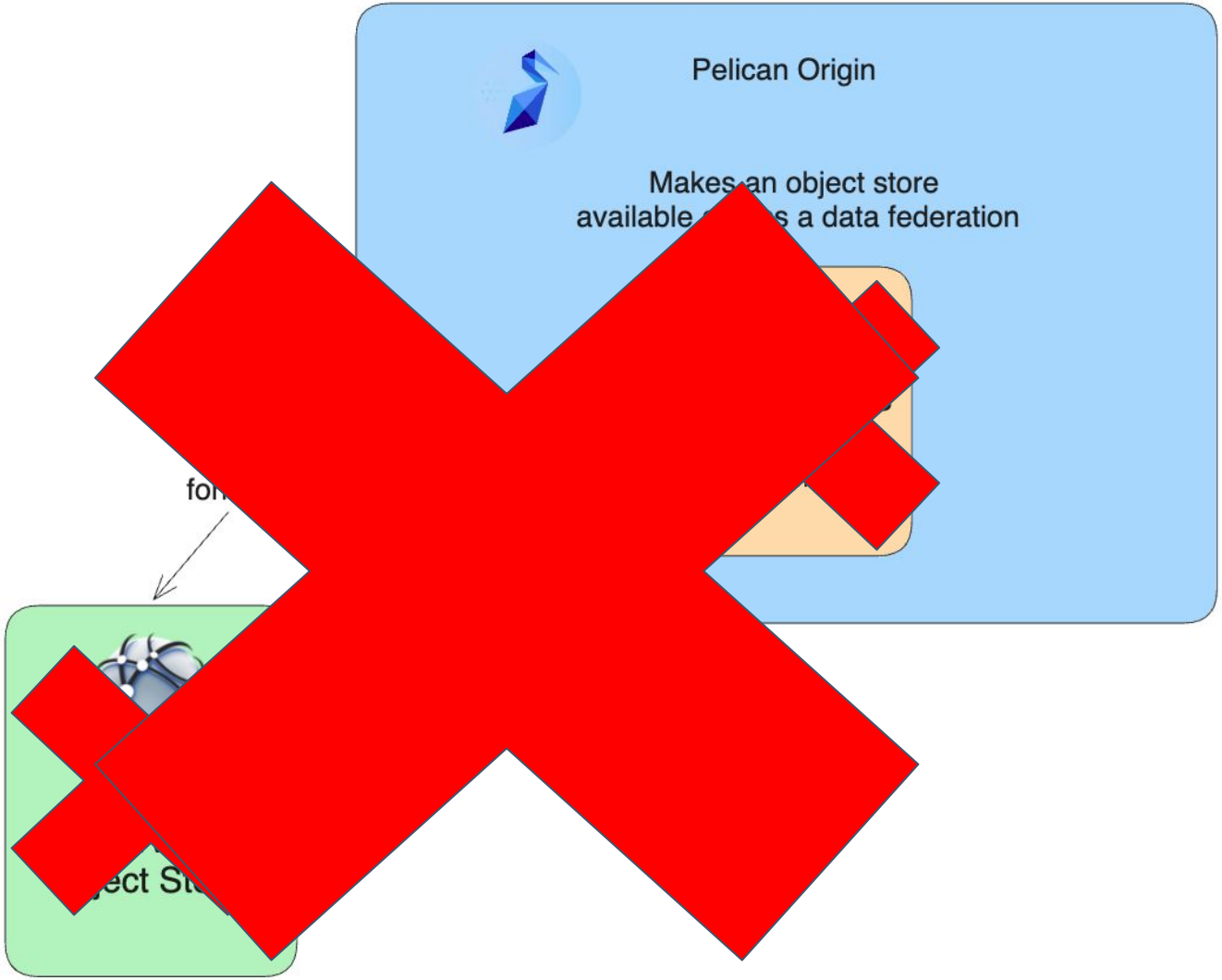
London

Lucky

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Problem: Diagnosing an Origin's Health

- Pelican Origins are the backbone to a data federation
  - Connects and Serves an object store
- Origins are critical within a federation, no origins -> no data
- If an origin goes down so does the data that it was serving
  - A staging device can't stage data if it can't access an origin
- Diagnosing why an origin is failing is difficult with the current tooling


Pelican Origin
Makes an object store
available
s a data federation

Status

Director
Failed to advertise to the director. Tests are not expected

Federation
XRootD server advertise failed: Director endpoint URL is not known

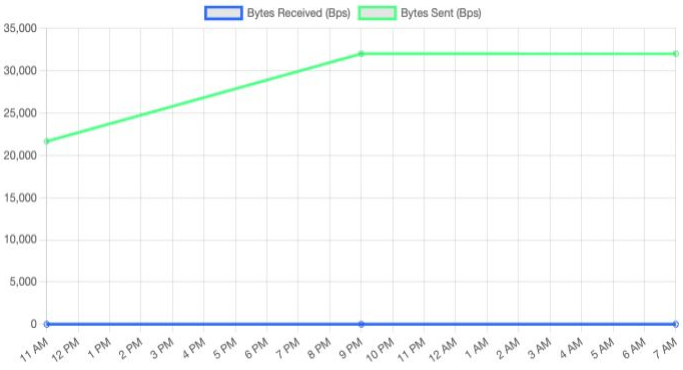
Registry

Web UI

XRootD

Last Updated: Jun 28, 2024, 9:40 AM

Transfer Rate



REPORT PERIOD | GRAPH SETTINGS

Rate Time Range: 28 | Unit: Hour

Resolution: 10 | Unit: Hour

Data Exports

Origin

Registration Completed

Namespaces

Federation Prefix
/patrickbrophy/test/origin3

Storage Prefix
/tmp/pelican

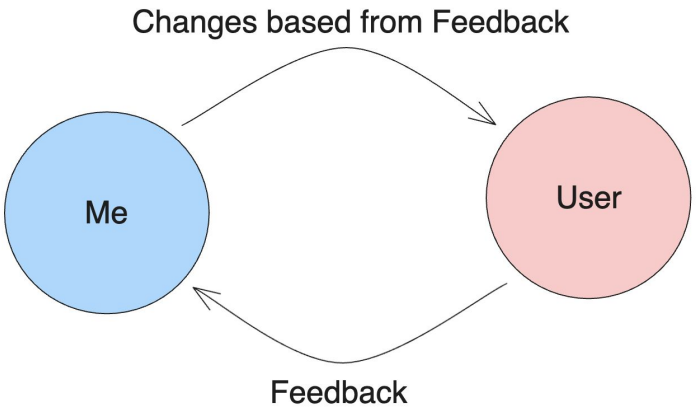
PublicRead	✓
Read	✓
Write	✗
Listing	✗
FallBackRead	✗

Federation Overview

- Registry
- Topology Namespace
- Discovery

Improving the Dashboard

- I conducted a user study with several Pelican system admins from CHTC and OSDF
- Feedback from system admins guided design choices



Origin Overview

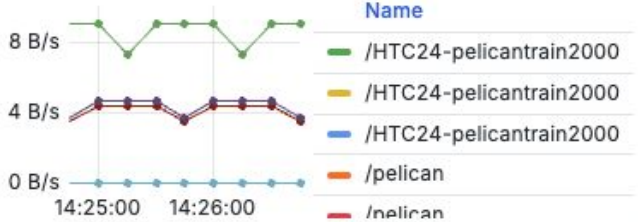
Component Status

director	federation	registry	web-ui	xrootd
Critical	Critical	OK	OK	OK

Server Status

Error

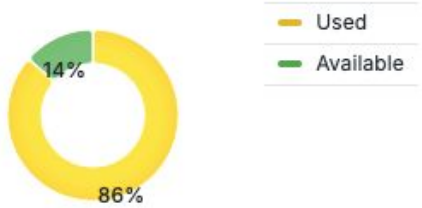
Transfer Rate



Top 5 Projects by Bytes Accessed

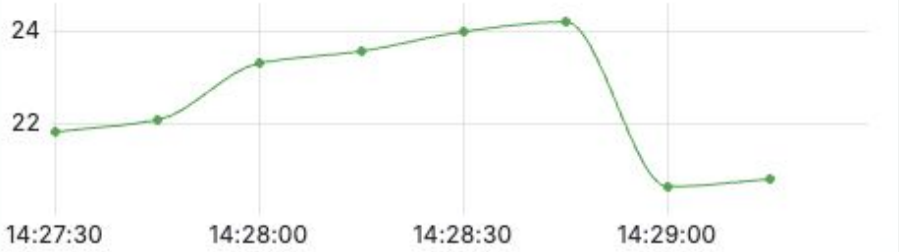
proj	Bytes Accessed
curl/8.6.0	5 B

Storage Usage



Resource Utilization

Memory (MB)



CPU Usage

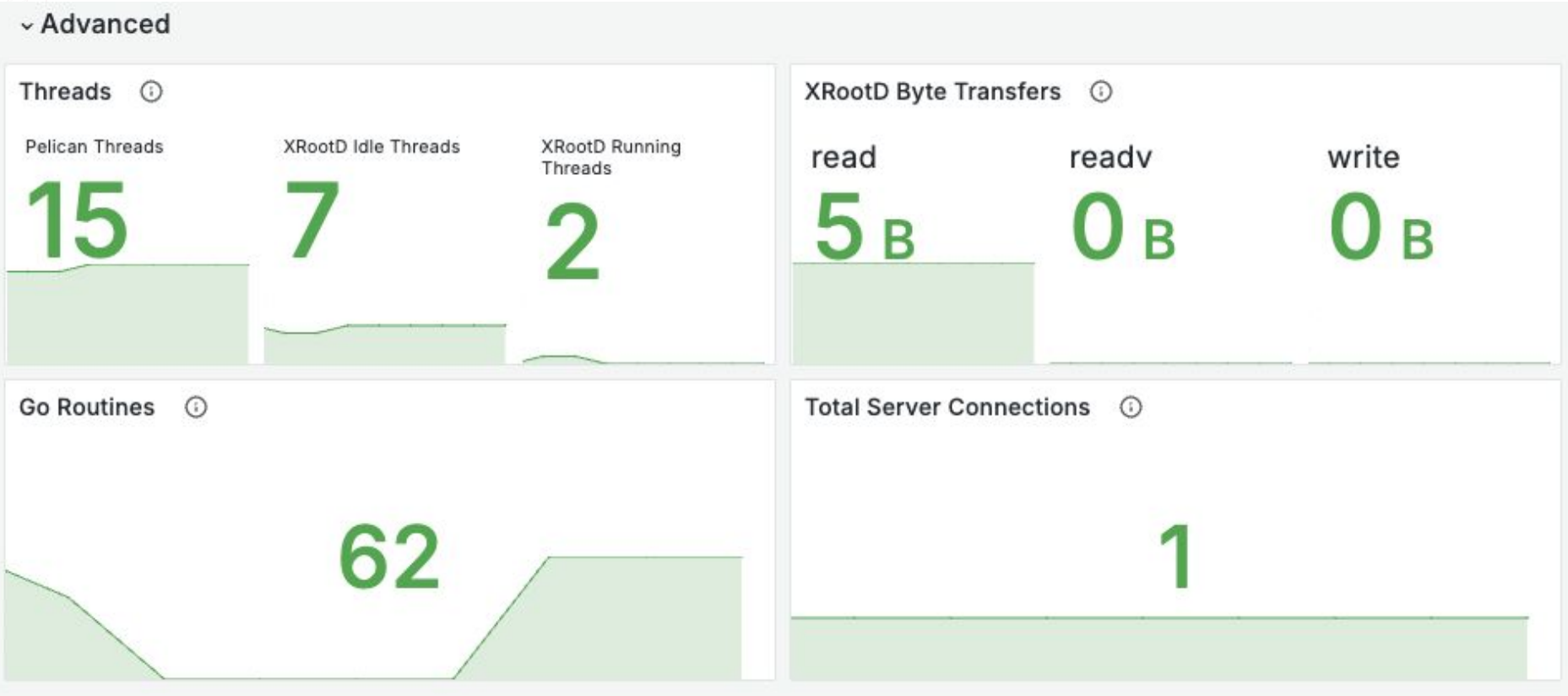
Total Bytes Transferred

Recieved

571 B

Transmitted

1.62 kB


Advanced
Threads
Pelican Threads
15
XRootD Idle Threads
7
XRootD Running Threads
2
XRootD Byte Transfers
read
5 B
readv
0 B
write
0 B
Go Routines
62
Total Server Connections
1


Memory (MB)
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
12:25
12:30
12:35
12:40
12:45
12:50
12:55
13:00
13:05
13:10
13:15

What's Next?

- More Metrics!
  - XRootD protocol-level (HTTP) metrics
- Alerting users of issues such as outages or warnings
- Reporting performance metrics for some period of time (day, week, month)
  - Staging Device miss rate
  - # of Objects accessed
  - Total bytes transferred

Thank you!

Please come and ask me questions!

CHTC Center For High Throughput Computing

Integrating Pelican with Pytorch

CHTC Summer Fellow: Kristina Zhao

Mentor: Ian Ross, Emma Turetsky

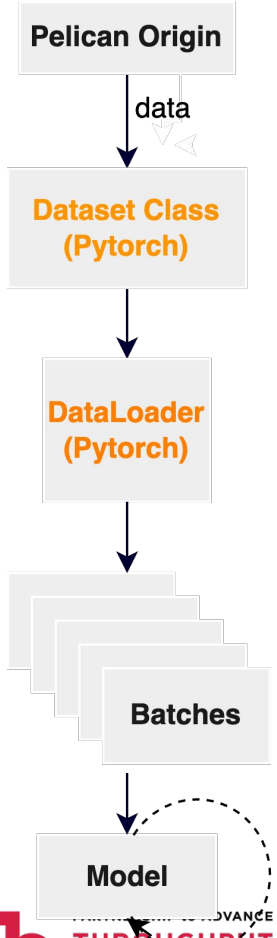
CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Introduction

Pelican 

PyTorch 

HTCondor 



CHTC PATh THROUGHPUT COMPUTING 

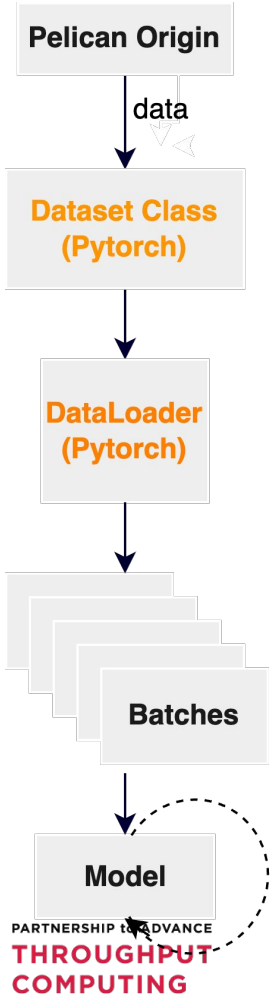
Problem

Data Accessibility

- Large Size
- Remote

Performance:

- low metadata latency
- high data throughput



Goals:

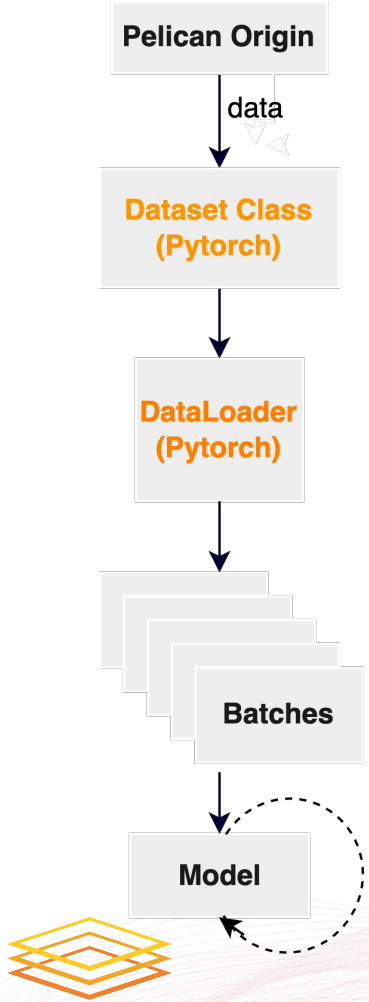
Streamlined Workflows

CLI -> Pelicanfs (implement fsspec)

Smoother integration

Efficient Data Handling

Make our AI Researchers happier 😄



CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Methodology

Research
- Pelican, fsspec, Pytorch data flows and requirements
- File format, file size, resource, limitation…

Benchmark

Develop tools/libraries

Tutorial and Documentation

Methodology

Research on Pelican and PyT

Benchmark

- local
- Pelicanfs
- Pelicanfs+Local Cache
- Pelicanfs+zip file

Develop tools/libraries

Tutorial and Documentation

Read from Local

In [10]:

```
s_time = time.time()

train_csv = pd.read_csv("input/fashion-mnist_train.csv")
test_csv = pd.read_csv("input/fashion-mnist_test.csv")

train_set = FashionDataset(train_csv, transform=transforms.Compose([transforms.ToTensor()]))
test_set = FashionDataset(test_csv, transform=transforms.Compose([transforms.ToTensor()]))

train_loader = DataLoader(train_set, batch_size=100)
test_loader = DataLoader(train_set, batch_size=100)

e_time = time.time()
print("Reading data time: ", e_time-s_time )

training()
```

```
Reading data time:  3.7035861015319824
Time of 1/3 epoch: 188.05s.
Time of 2/3 epoch: 182.10s.
Time of 3/3 epoch: 188.48s.
```

Read from Pelican using Pelicanfs

In [9]:

```
s_time = time.time()
fs = PelicanFileSystem("pelican://osg-htc.org")
train_csv = pd.read_csv(fs.open('/chtc/PUBLIC/hzhao292/fashion-mnist_train.csv', 'rb'))
test_csv = pd.read_csv(fs.open('/chtc/PUBLIC/hzhao292/fashion-mnist_test.csv', 'rb'))

train_set = FashionDataset(train_csv, transform=transforms.Compose([transforms.ToTensor()]))
test_set = FashionDataset(test_csv, transform=transforms.Compose([transforms.ToTensor()]))

train_loader = DataLoader(train_set, batch_size=100)
test_loader = DataLoader(train_set, batch_size=100)
e_time = time.time()
print("Reading data time: ", e_time-s_time )

training()
```

```
Reading data time:  13.530436038970947
Time of 1/3 epoch: 183.30s.
Time of 2/3 epoch: 250.75s.
Time of 3/3 epoch: 192.42s.
```

Methodology

Research on Pelican and F

Benchmark

Develop tools/libraries

- Pelicanfs
- Pelican connector?

Tutorial and Documentatio

CHTC F

ImageFolder

CLASS torchvision.datasets.ImageFolder(*root: str, transform: ~typing.Optional[~typing.Callable] = None, target_transform: ~typing.Optional[~typing.Callable] = None, loader: ~typing.Callable[[str], ~typing.Any] = <function default_loader>, is_valid_file: ~typing.Optional[~typing.Callable[[str], bool]] = None, allow_empty: bool = False*) [SOURCE]

A generic data loader where the images are arranged in this way by default:

```
root/dog/xxx.png
root/dog/xxy.png
root/dog/[...]/xxz.png

root/cat/123.png
root/cat/nsdf3.png
root/cat/[...]/asd932_.png
```

This class inherits from `DatasetFolder` so the same methods can be overridden to customize the dataset.

Parameters:

- **root** (str or `pathlib.Path`) – Root directory path.
- **transform** (*callable, optional*) – A function/transform that takes in a PIL image and returns a transformed version. E.g, `transforms.RandomCrop`
- **target_transform** (*callable, optional*) – A function/transform that takes in the target and transforms it.
- **loader** (*callable, optional*) – A function to load an image given its path.
- **is_valid_file** (*callable, optional*) – A function that takes path of an Image file and check if the file is a valid file (used to check of corrupt files)
- **allow_empty** – If True, empty folders are considered to be valid classes. An error is raised on empty folders if False (default).

Methodology

Research on Pelican and PyTorch data needs

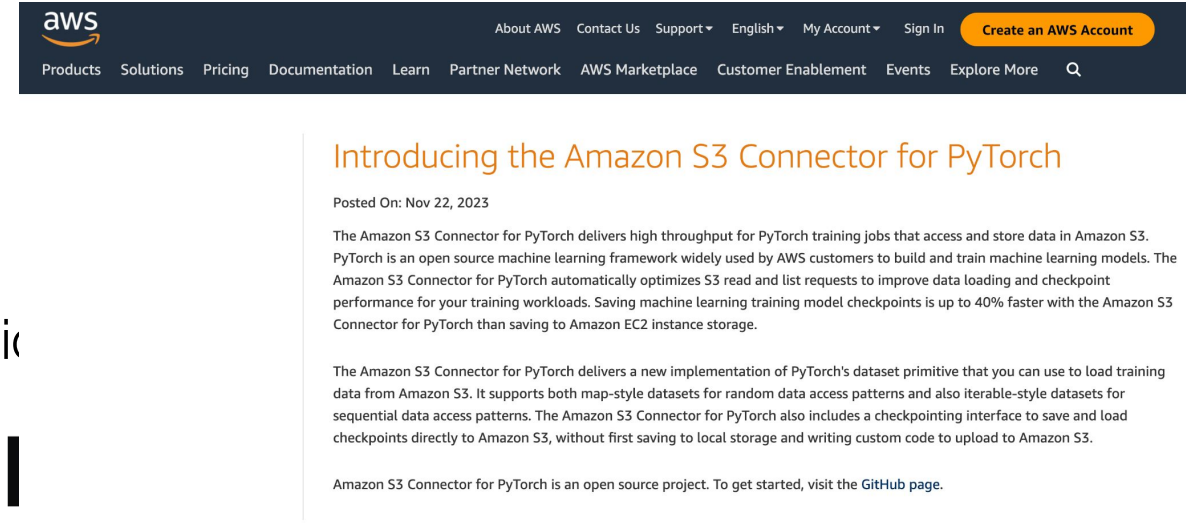
Benchmark

Develop tools/libraries

- Pelicanfs
- Pelican connector?

Tutorial and Documentati

CHTC



Methodology

Research on Pelican and PyTorch data needs

Benchmark

Develop tools/libraries

Tutorial and Documentatic

Thank you!

Discussion and Problems welcome!

Kristina Zhao
hzhao292@wisc.edu

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Enhancing the Building of the OSG Container Images

Pratham Patel

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Who Am I?

- Hometown: Beloit, WI
- Senior at UW-Madison studying CS & DS
- I'm pretty awesome…

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Enhancing the OSG Container Build System

A Three-Phase Approach to Versatility and Efficiency

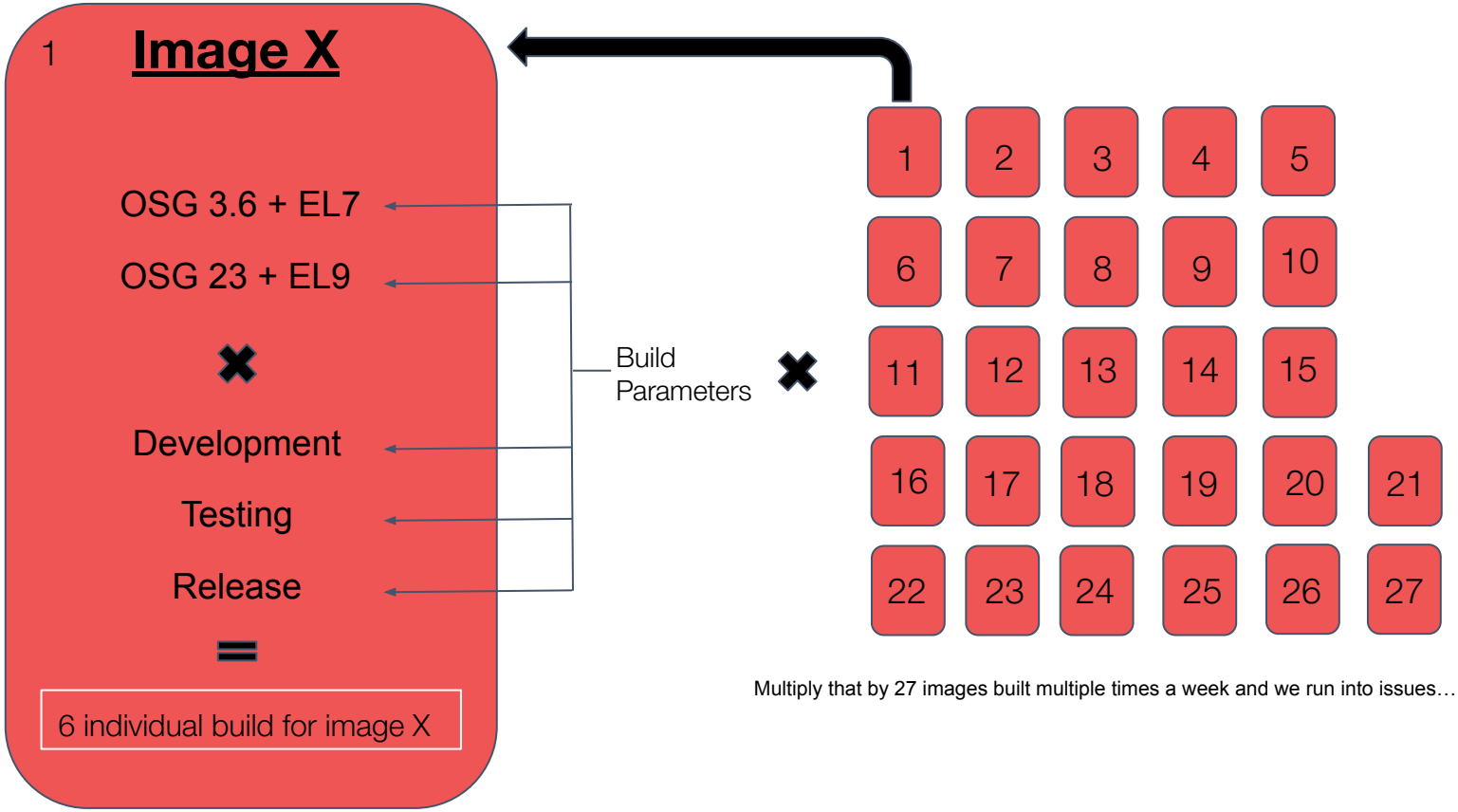
Abstract:

- CHTC builds images for sites to run and for use internally.
- Images are based on upstream OS container images
- We build all images at least once a week

Focus:

- Adding versatility and streamlining the build process within the OSG images repository.
- Three-phase approach:
  1. Customizable build instructions for each image.
  2. Dynamic trigger for external repositories.
  3. Compatibility and support for ARM-based systems.

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING



Multiply that by 27 images built multiple times a week and we run into issues…

Background

Current State:

- **GitHub Actions Workflow:**
  - Automates building and pushing container images.
  - Triggered by specific conditions and events.
- **Monolithic Design:**
  - Lack of flexibility in the build process.
  - All images built using a single, unified workflow.
- **Additional Features:**
  - Not supported for ARM architecture.

Project Requirements

Versatility:

- Mechanism for custom build processes through a unique configuration file.
- Default instructions in absence of unique configuration file.

Trigger Mechanism:

- Located in the images repository.
- Activates updates with Pelican and other external repositories.

ARM Compatibility:

- Add support for building native ARM-based systems.
- Aim for ARM-optimized Pelican images.

Solution

Phase 1: Customizable Build Instructions

- **Objective:** Establish an advanced image repository framework with configurable build instructions.
- **Implementation Paths:**
  - Dynamic parameters for building images.
  - Modularize workflow into reusable components.

Phase 2: Triggering the Pelican Repository

- **Objective:** Integrate with the external repositories using a trigger.
- **Implementation Paths:**
  - Create some trigger in the images repository.
  - Use GitHub Actions to monitor and trigger Pelican repository and others.
  - Develop GitHub Action in Pelican repository to handle the trigger for updates.

Conclusion

Summary:

- Implementing a three-phase approach that enhances flexibility, efficiency, and future-readiness.
- Custom build instructions, trigger for external repositories like Pelican, and ARM-based support.

Next Steps:

- Refine customizable build instructions
- Work with Pelican team to determine trigger strategies

CHTC Fellowship: Tracking Server Inventory and Elevation

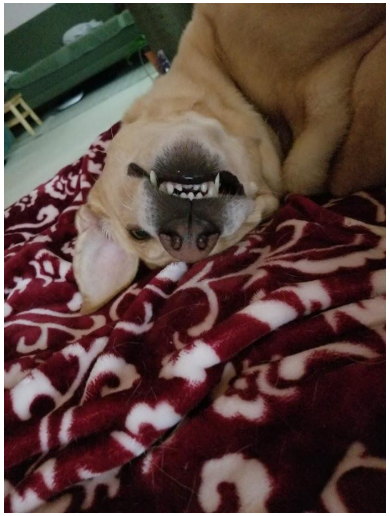
Ben Staehle

CHTC

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Who am I?

- My name is Ben Staehle
- Madison native - currently at UW-Madison
- This summer - CHTC Fellowship
- Working with Joe Bartkowiak on
  “Tracking Server Inventory and Elevation”



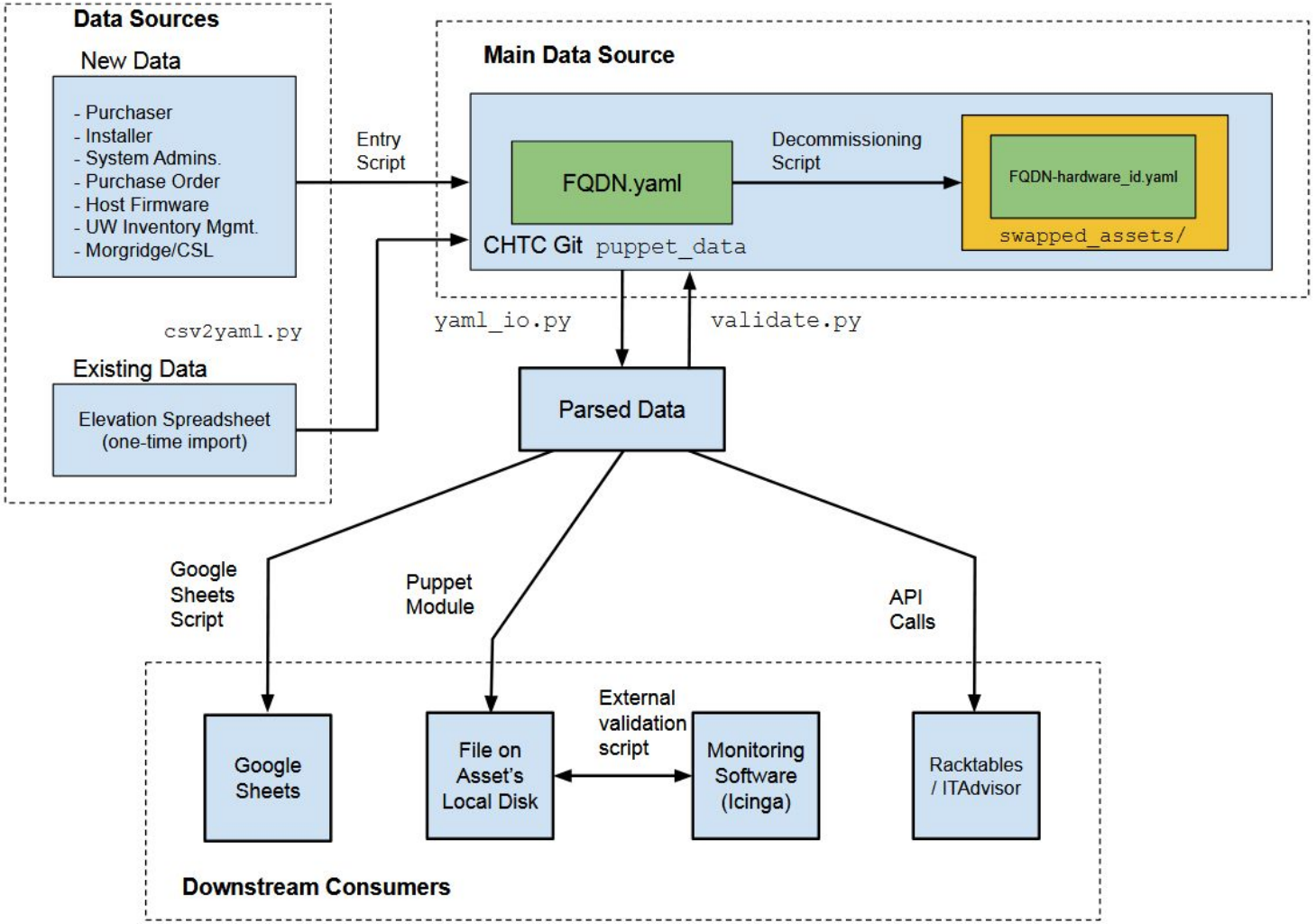
Willow

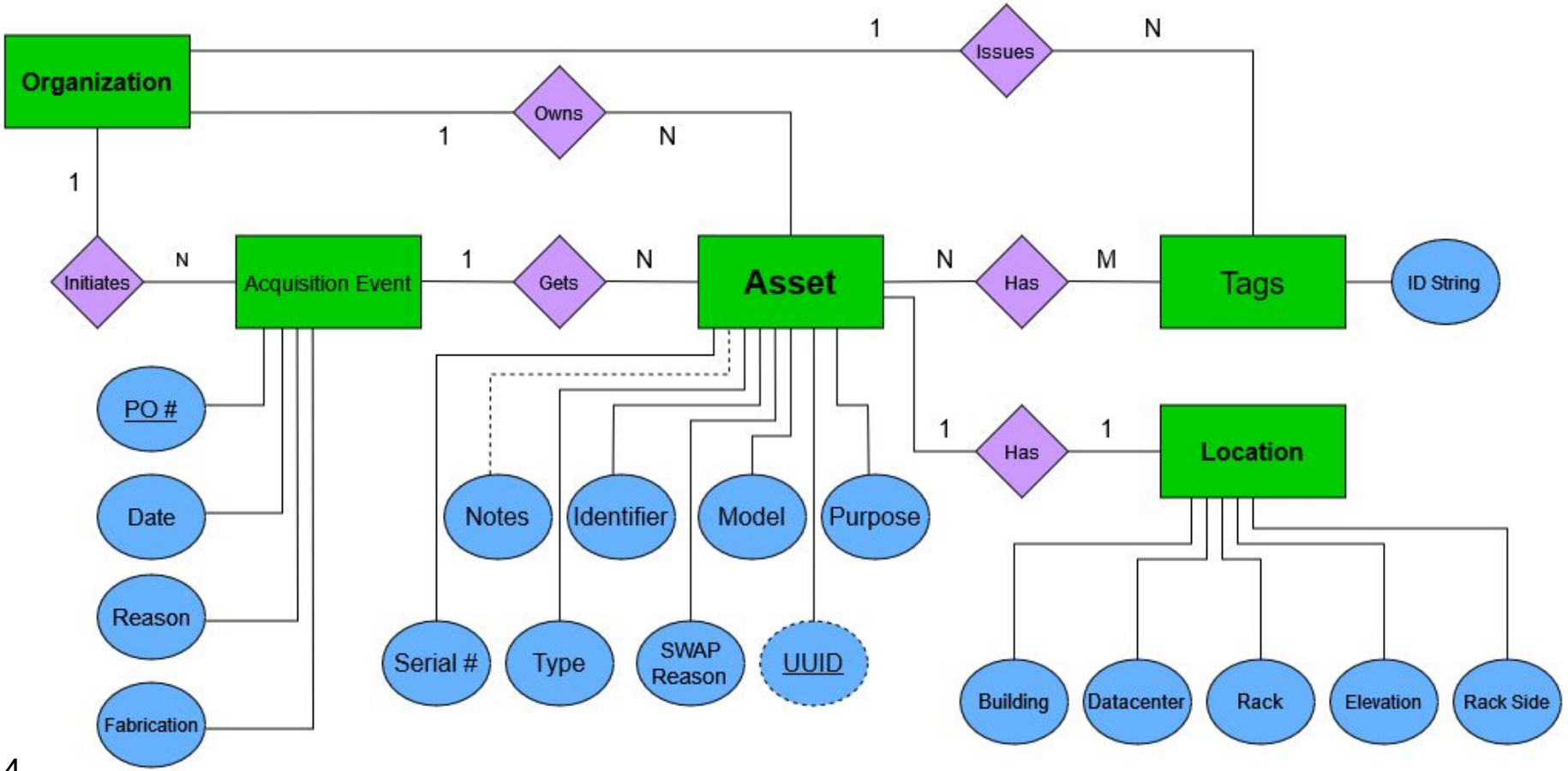
Me

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

What are we solving?

- CHTC maintains more than 1200 assets
- System administrators are responsible for maintaining inventory records
- Interested stakeholders include UW-Madison, Morgridge Institute for Research
- Previous internal asset tracking was cumbersome


Data Sources
New Data
- Purchaser
- Installer
- System Admins.
- Purchase Order
- Host Firmware
- UW Inventory Mgmt.
- Morgridge/CSL
csv2yaml.py
Existing Data
Elevation Spreadsheet (one-time import)
Entry Script
Main Data Source
FQDN.yaml
Decommissioning Script
FQDN-hardware_id.yaml
swapped_assets/
CHTC Git puppet_data
yaml_io.py
validate.py
Parsed Data
Google Sheets Script
Puppet Module
API Calls
Google Sheets
File on Asset's Local Disk
External validation script
Monitoring Software (Icinga)
Racktables / ITAdvisor
Downstream Consumers


Organization
1
Issues
N
1
Owns
N
1
Initiates
N
Acquisition Event
1
Gets
N
Asset
N
Has
M
Tags
ID String
PO #
Date
Reason
Fabrication
Notes
Identifier
Model
Purpose
Serial #
Type
SWAP Reason
UUID
1
Has
1
Location
Building
Datacenter
Rack
Elevation
Rack Side

Thank you!

Questions?

Performance Monitoring in the Schedd

Wil Cram

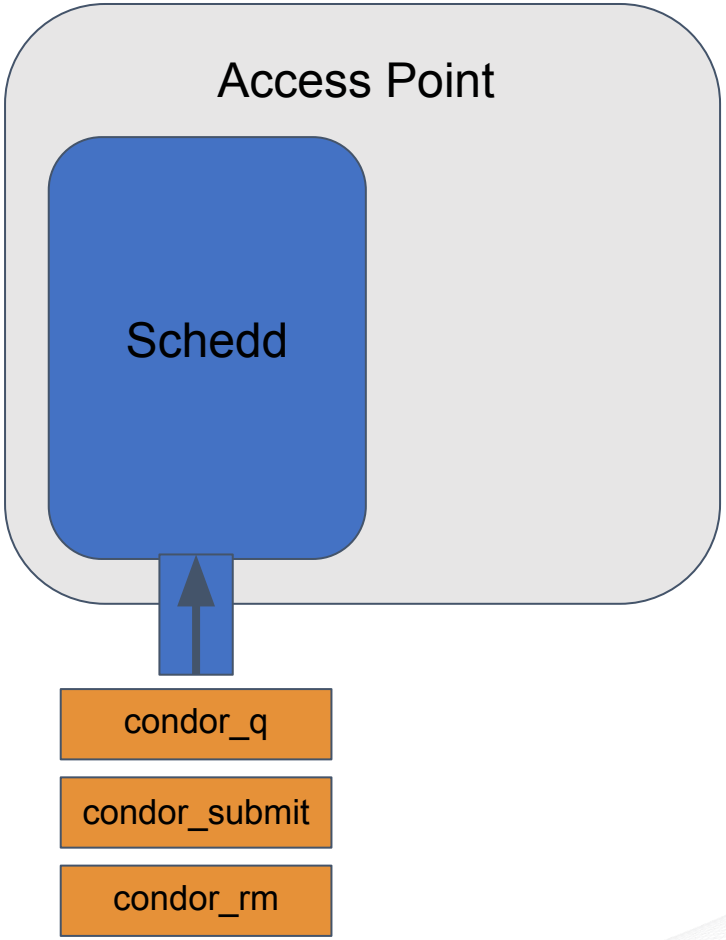
Greg Thain

CHTC

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

The Motivation

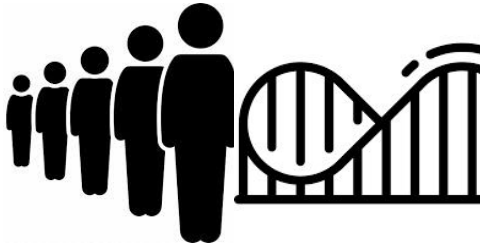
- The Schedd process handles commands
- Sometimes it slows down without warning
- How to find the root cause?
- Admins: who is the problem?



Definition 1.1

A *task* is an object with an associated "time cost"

A *queue* is an object that accepts incoming tasks and works on them one at a time



The Problem

- Consider a queue Q with categories 1,2,…,n
- The queue records the total time spent processing each category
- How do we determine real-time data about this system?

Enter Sampling

- We can record the totals every x seconds
- The time spent over some window is `new_runtime - old_runtime`
- We can divide by the sum of totals to get a relative proportion
- This tells us what is taking up time in the queue

The Schedd is a Queue!

(Basically)

CHTC

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

The Schedd:

- Operates on tasks one at a time
- Has categories
  (commands, handling jobs, etc.)
- Publishes totals through ClassAds

Trivial, Right?

```
2024-06-28 15:32:04----------------------------------------------------------
Sample Window:                5 s
CPU Time:                 0.088 s  <- How???
CPU Usage:                 1.77%
------------------------------------------------------------------------------
 Commands (0.66%)
  * wil@fourier: 0.34%
    - command_query_job_ads: 0.31%
    - command_query_ads: 0.04%
  * test@fourier: 0.31%
    - command_query_job_ads: 0.31%
 Other (1.11%)
------------------------------------------------------------------------------
Waiting 2 seconds
```

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

Wrong!

- Condor publishes the average % of time that CPU is recently “busy”(20min window)
- Multiply by window size to get CPU time?
- If the percentage doesn’t move much, sure

```
2024-06-28 15:42:00----------------------------------------------------------
Sample Window:        5 s
CPU Time:         1.268 s
CPU Usage:         25.35%
-------------------------------------------------------------------------------
 Commands (14.37%)
  * test@fourier: 10.31%
    - handle_q: 10.29%
    - reschedule_negotiator: 0.02%
  * wil@fourier: 2.53%
    - command_query_job_ads: 2.44%
    - command_query_ads: 0.09%
  * condor@child: 1.53%
    - handle_q: 1.52%
 TotalFsync (3.27%)
 Timers (0.97%)
  * StartJobHandlerRuntime: 0.47%
  * timeoutRuntime: 0.27%
  * SelfDrainingQueuetimerHandlerjob_is_finished_queueRuntime: 0.15%
  * checkContactQueueRuntime: 0.08%
 Other (6.74%)
-------------------------------------------------------------------------------
```

Wrong!

- Condor publishes the average % of time that CPU is recently “busy”(20min window)
- Multiply by window size to get CPU time?
- If the percentage doesn’t move much, sure
- This average falls behind during a traffic spike

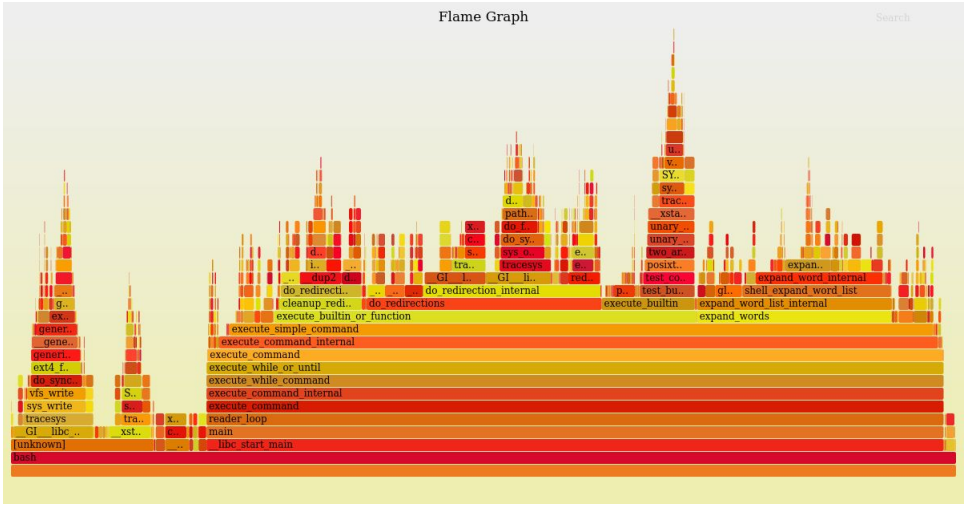
```
2024-06-28 15:44:56----------------------------------------------------------
Sample Window:        5 s
CPU Time:         1.834 s
CPU Usage:         36.68%
-------------------------------------------------------------------------------
 WARNING: Summed statistics exceeded CPU time! These values are renormalized!
 This typically happends when traffic spikes faster then CPU Usage can update.
 Commands (25.52%)
  * test@fourier: 13.73%
    - handle_q: 13.72%
    - reschedule_negotiator: 0.02%
  * condor@child: 11.76%
    - handle_q: 11.76%
  * wil@fourier: 0.02%
    - command_query_ads: 0.02%
 TotalFsync (9.79%)
 Timers (1.37%)
  * SelfDrainingQueuetimerHandlerjob_is_finished_queueRuntime: 0.78%
  * timeoutRuntime: 0.59%
 Other (0.00%)
-------------------------------------------------------------------------------
Waiting 4 seconds
```

Wrong!

- Condor publishes the average % of time that CPU is recently “busy”(20min window)
- Multiply by window size to get CPU time?
- If the percentage doesn’t move much, sure
- This average falls behind during a traffic spike
- We trust our own recent totals
- However, we lose information

Future Plans

- Improving visibility at a glance
- More process coverage
- Give intelligent warnings to users



"User condor_q_spammer has taken over 25% of busy time for the past hour. Maybe don't let them do that?"

CHTC PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING