# Slurm's Rest API

Mentors: David Fox, Ryan Day
Wesley Hsieh
Fnu Azma

August 11, 2021

LLNL-PRES-825699

# A brief Introduction

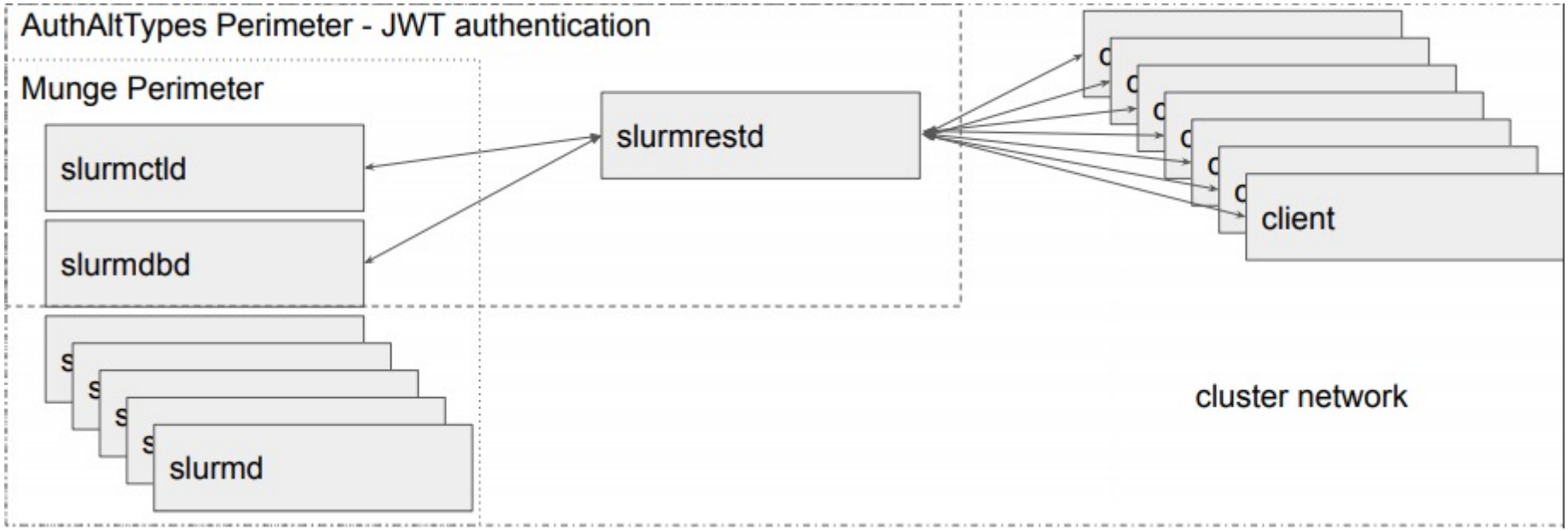
- Wesley Hsieh
- Senior at CSUEB
- Computer Science
- Expected Grad: Dec 2021

- Fnu Azma
- Junior at UCR
- Computer Science and Engineering
- Expected Grad: Dec 2022

# Slurm and Slurm's Rest API

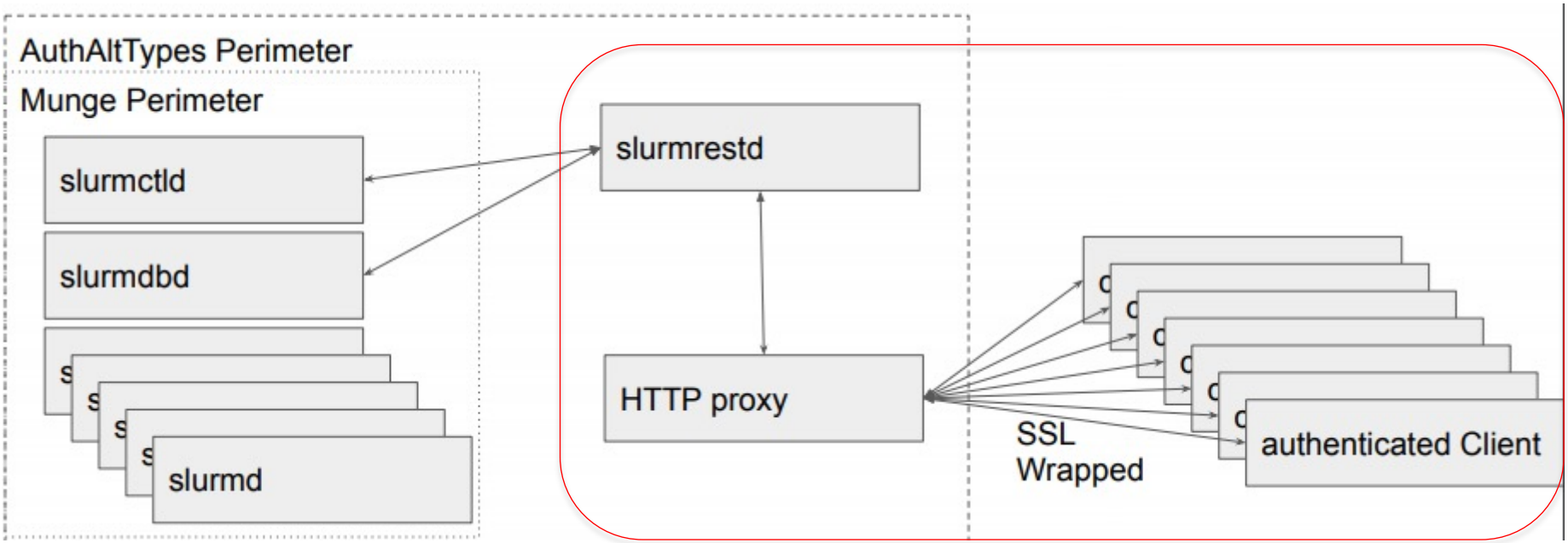
- Slurm:
  - Job scheduler for Linux and Unix systems
  - Features: centralized manager (slurmctld), the executors (slurmd), an "accounting database" (slurmdbd), and it's own REST API (slurmrestd).
- "A tool that runs inside of the Slurm perimeter that will translate JSON/YAML requests into Slurm RPC requests."
  - Authentication via Http headers: X-SLURM-USER-TOKEN (auth/jwt)  X-SLURM-USER-NAME

# Slurmrestd Architecture




Courtesy of https://slurm.schedmd.com/PEARC20/REST_API.pdf

# HTTP Proxy Front End




Courtesy of https://slurm.schedmd.com/PEARC20/REST_API.pdf

# Project Objectives

- Enable Slurm REST API on management nodes
- Slurm REST API – explore sample code, implement in python.
- Configure/enable the use of a proxy server (NGINX) as an added layer of security.

# Slurm REST Calls

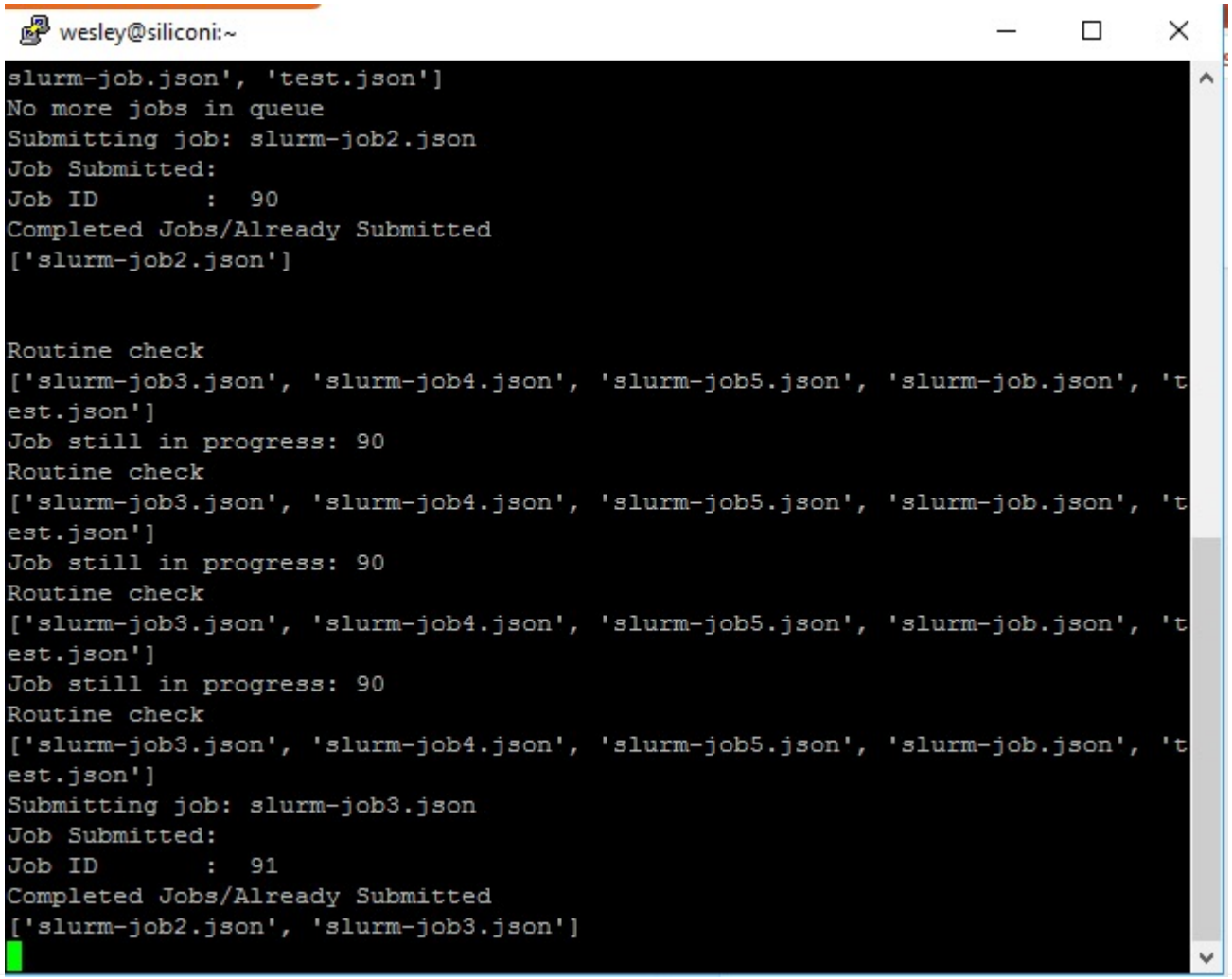
- DELETE /slurm/v0.0.36/job/{job_id}
- GET /slurm/v0.0.36/diag
- GET /slurm/v0.0.36/job/{job_id}
- GET /slurm/v0.0.36/jobs
- GET /slurm/v0.0.36/node/{node_name}
- GET /slurm/v0.0.36/nodes
- GET /slurm/v0.0.36/partition/{partition_name}
- GET /slurm/v0.0.36/partitions
- GET /slurm/v0.0.36/ping
- POST /slurm/v0.0.36/job/submit
- POST /slurm/v0.0.36/job/{job_id}
- POST /slurmdb/v0.0.36/clusters
- POST /slurmdb/v0.0.36/wckeys
- DELETE /slurmdb/v0.0.36/account/{account_name}

https://slurm.schedmd.com/rest_api.html

# What we actually achieved

- Documenting key aspects of the installation process specific to our clusters.
- Basic python script/example code of utilizing REST API calls.
- Basic example of web proxying via NginX

# “Producer-Consumer” Python Script



# Example Job file (JSON)

```
{
"jobs": {
"tasks": 1,
"name": "test1",
"nodes": 4,
"current_working_directory": "/home/wesley",
"environment": {"PATH": "/bin:/usr/bin/:/usr/local/bin/",
"LD_LIBRARY_PATH":"/lib/:/lib64/:/usr/local/lib"}},
"script":"#!/bin/bash\n sleep 15"}
~
~
```

# Example of job submission:

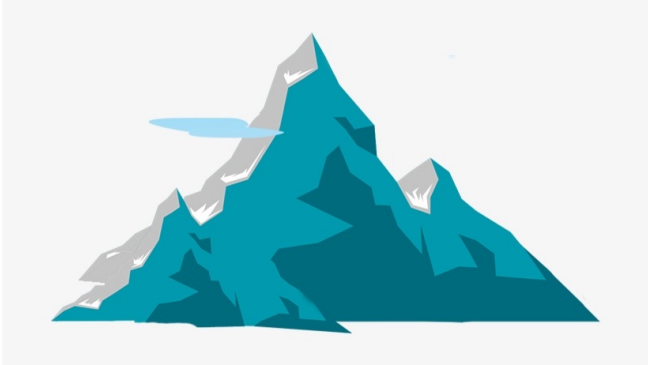
```
[wesley@siliconi ~]$ curl -H "X-SLURM-USER-NAME:$(whoami)" -H "X-SLURM-USER-TOKE
N:$SLURM_JWT" -X POST http://127.0.0.1:5432/slurm/v0.0.36/job/submit -H "Content
-Type: application/json" -d @slurm-job.json
{
   "meta": {
     "plugin": {
       "type": "openapi\/v0.0.36",
       "name": "REST v0.0.36"
     },
     "Slurm": {
       "version": {
         "major": 20,
         "micro": 7,
         "minor": 11
       },
       "release": "20.11.7"
     }
   },
   "errors": [
   ],
   "job_id": 96,
   "step_id": "BATCH",
   "job_submit_user_msg": ""
 }[wesley@siliconi ~]$
```

# Example of NginX functionality

```
[wesley@siliconi ~]$ curl -H "X-SLURM-USER-NAME:$(whoami)" -H "X-SLURM-USER-TOKE
N:$SLURM_JWT" http://192.168.95.1:8090/slurm/v0.0.36/ping
{
   "meta": {
     "plugin": {
       "type": "openapi\/v0.0.36",
       "name": "REST v0.0.36"
     },
     "Slurm": {
       "version": {
         "major": 20,
         "micro": 7,
         "minor": 11
       },
       "release": "20.11.7"
     }
   },
   "errors": [
   ],
   "pings": [
     {
       "hostname": "siliconi",
       "ping": "UP",
       "status": 0,
       "mode": "primary"
     }
   ]
```

# Some Challenges

- Limited web resources
  - A lot of “trial and error” with API calls due to unclear documentation
  - Trusty old `tail /var/log/slurm/slurmctld.log`
- “High Barrier to Entry”
- Lots of command line usage: i.e. vim, curl, tar



# Future Improvements, “Where to go from here”

- Configure Slurm’s database to work with slurmrestd
- Running slurmrestd in the background
  - `systemctl start slurmrestd` vs.
  - `slurmrestd -f /etc/slurm/slurmrestd.conf -s openapi/v0.0.36 -vvvvv 127.0.0.1:[port number]`
- Possible considerations to a more fleshed out web proxy service using NginX/Apache
  - Web Application with two-factor authentication (i.e. RSA-token, AD-native authentication)

# Citations/Resources

- https://slurm.schedmd.com/rest_api.html
- https://slurm.schedmd.com/rest.html
- https://nginx.org/en/docs/
- https://www.youtube.com/watch?v=RtdJlstFB28
- https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uswgi-and-nginx-on-ubuntu-18-04
- https://www.programmersought.com/article/48456629330/

# Disclaimer and logo