

A RESTful workbench of the Slurm cluster at IHEP

Ran Du duan@ihep.ac.cn

On behalf of the scheduling group, IHEP-CC

2022-03-22

Outline

- Background & Motivation
- Slurm REST APIs
- Development of the Workbench
- Conclusion and Next

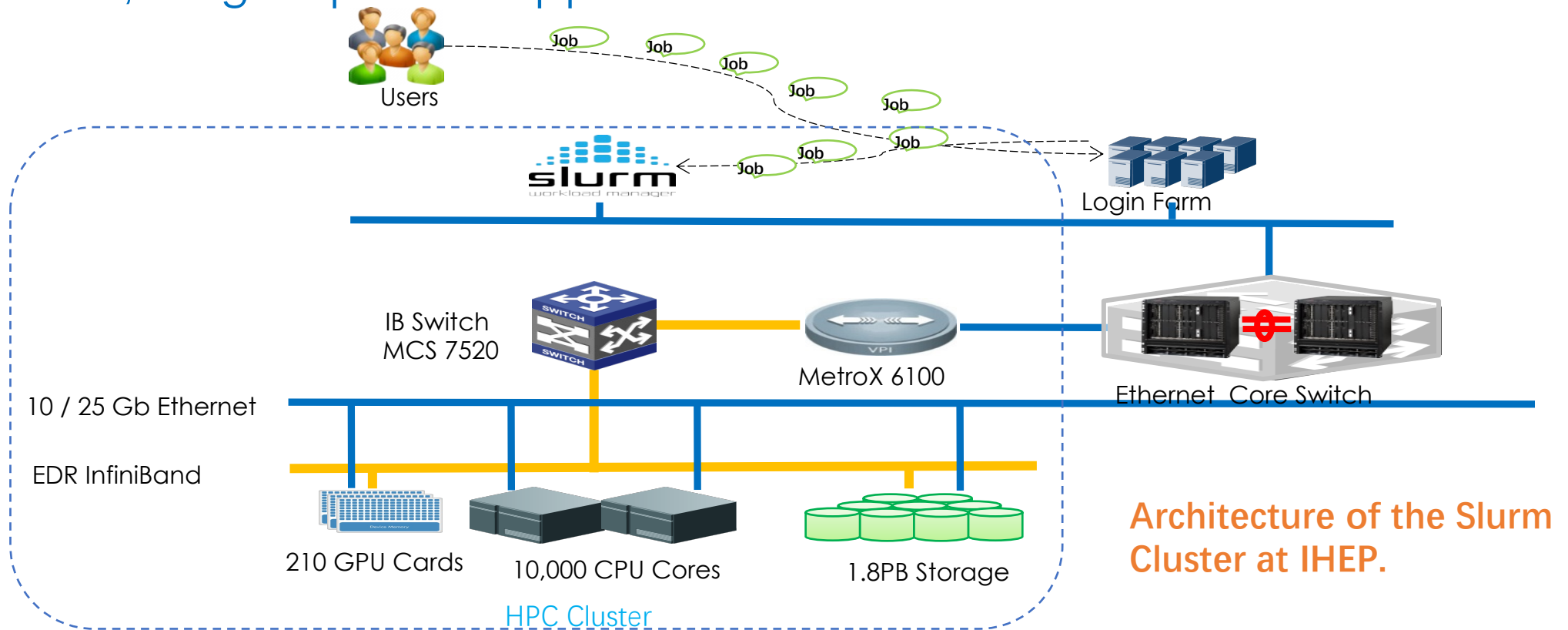


Background & Motivation



The Slurm Cluster at IHEP

- To run parallel and heterogeneous jobs.
- 228 worker nodes, 10K CPU cores, 210 GPU cards.
- 480 users, 20 groups are supported.



Why is the RESTful workbench needed?

- Motivation

- Effective maintenance
- Research purpose
- Manpower in shortage
- Multiple application systems under a unified framework



Slurm REST APIs



What is REST

- REST : REpresentational State Transfer
- An architectural style for distributed hypermedia systems
- First presented by Roy Fielding in 2000
- Guiding Principles of REST
 - Client–server
 - Stateless
 - Cacheable
 - Uniform interface
 - Layered system
 - Code on demand (optional)

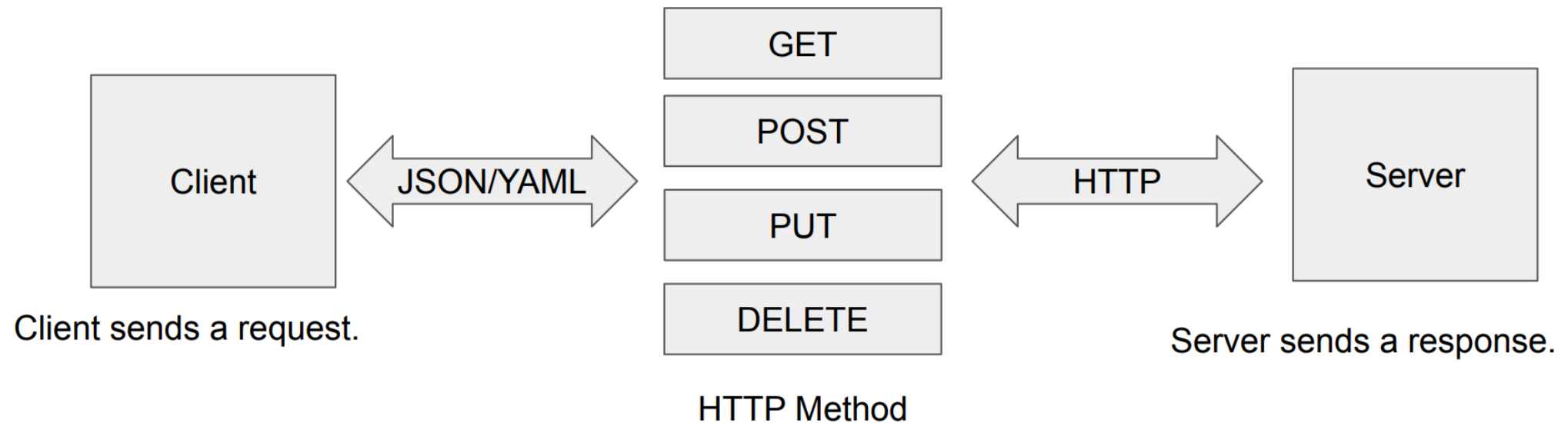
Reference : https://slurm.schedmd.com/SLUG20/REST_API.pdf



REST API Workflow

- Client-Server architecture
- URL as the unique resource ID
- Access with HTTP protocol

- Client-Server
- Stateless
- Cacheable
- Uniform interface
- Layered system

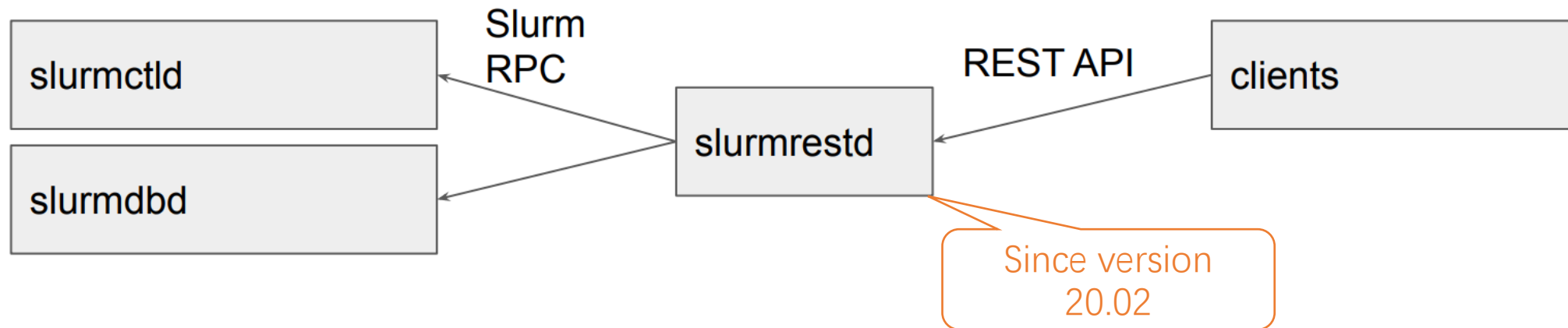


Reference : https://slurm.schedmd.com/SLUG20/REST_API.pdf



slurmrestd

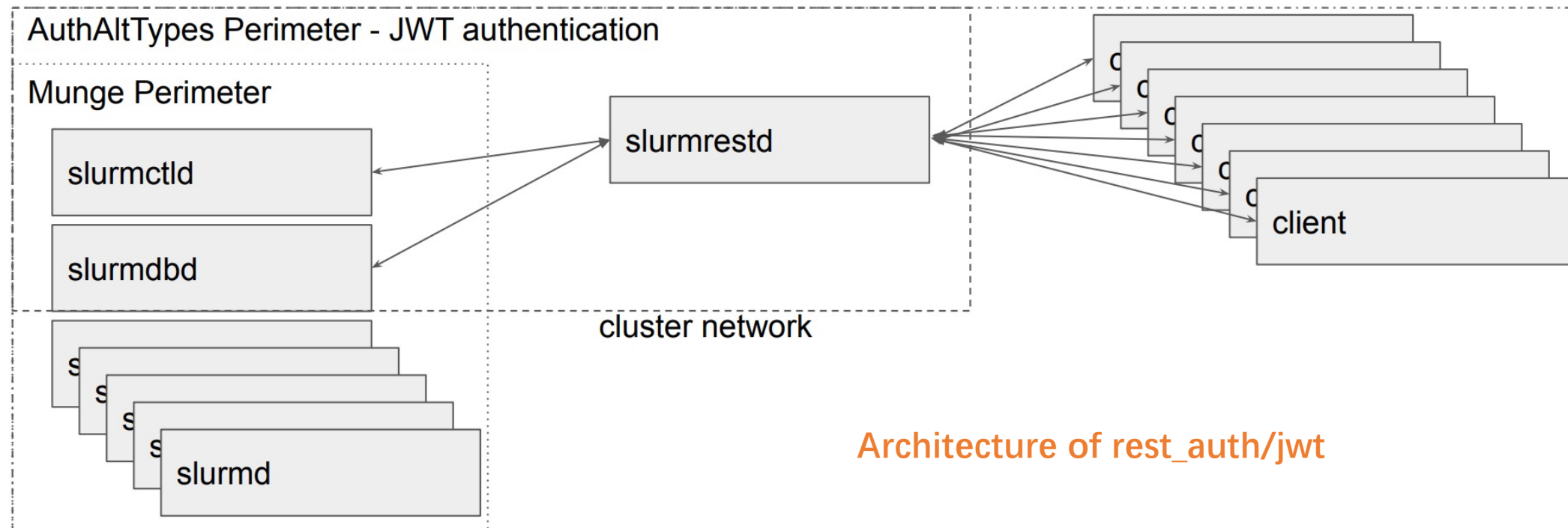
- Slurmrestd interacts with slurmctld & slurmdbd with Slurm RPC.
- RESTful APIs are provided to clients.
- Written in C.



Reference : https://slurm.schedmd.com/SLUG20/REST_API.pdf

Slurm REST API Architecture

- The API architecture used in our workbench : rest_auth/jwt
- The others are rest_auth/local, rest_auth/jwt + proxy



Reference : https://slurm.schedmd.com/SLUG20/REST_API.pdf

Slurm REST APIs

- Reference : [Slurm REST API Reference](#)
- GET, POST, DELETE methods are provided, no PUT methods.

slurmctld

- [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/wckey](#)

slurmdbd

- [GET /slurmdb/v0.0.36/accounts](#)
- [GET /slurmdb/v0.0.36/association](#)
- [GET /slurmdb/v0.0.36/associations](#)
- [GET /slurmdb/v0.0.36/cluster/{cluster_name}](#)
- [GET /slurmdb/v0.0.36/clusters](#)
- [GET /slurmdb/v0.0.36/config](#)
- [GET /slurmdb/v0.0.36/job/{job_id}](#)
- [GET /slurmdb/v0.0.36/jobs](#)
- [GET /slurmdb/v0.0.36/qos](#)
- [GET /slurmdb/v0.0.36/qos/{qos_name}](#)
- [GET /slurmdb/v0.0.36/tres](#)
- [GET /slurmdb/v0.0.36/user/{user_name}](#)
- [GET /slurmdb/v0.0.36/users](#)



Example of Slurm REST POST methods

Request

```
$ curl -H X-SLURM-USER-NAME:$(whoami) \
-H X-SLURM-USER-TOKEN:$SLURM_JWT \
-X POST 'http://slurm06.ihep.ac.cn:9999/slurm/v0.0.36/job/submit' \
-H "Content-Type: application/json" \
-d @submit_job_array.json
```

Username and JWT tokens
are mandatory

```
$ cat submit_job_array.json
{
  "job":{
    "account":"u07",
    "partition":"gpu",
    "qos":"regular",
    "array":"1-3",
    "ntasks":4,
    "memory_per_cpu":4096,
    "name":"job_array_test",
    "standard_out":"/tmp/rest_array_job_%j.out",
    "environment":{
      "PATH":"/bin:/usr/bin:/usr/local/bin",
      "LD_LIBRARY_PATH":"/lib:/lib64:/usr/local/lib"
    },
    "script":"#!/bin/bash\nsleep 120"
  }
}
```

Input data

Response

```
{
  "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": 67109296,
      "step_id": "BATCH",
      "job_submit_user_msg": ""
    }
  ]
}
```

Return a job
ID if successful.



Example of Slurm REST GET methods

Request

```
$ curl -H X-SLURM-USER-NAME:${whoami} \
-H X-SLURM-USER-TOKEN:${SLURM_JWT} \
-X GET 'http://localhost:9999/slurm/v0.0.36/jobs'
```

Username and JWT tokens
are mandatory

Return detailed job description if successful

Response

```
{
  "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": [
  ],
  "jobs": [
    {
      "account": "u07",
      "accrue_time": 1647255451,
      "admin_comment": "",
      "array_job_id": 0,
      "array_task_id": null,
      "array_max_tasks": 0,
      "array_task_string": "",
      "association_id": 4,
      "batch_features": "",
      "batch_flag": true,
      "batch_host": "slurm03",
      "flags": [
        "JOB_ACCRUE_OVER",
        "JOB_WAS_RUNNING",
        "JOB_MEM_SET"
      ],
      "burst_buffer": "",
      "burst_buffer_state": "",
      "cluster": "slurm_testbed_1",
      "cluster_features": "",
      "command": "",
      "comment": "",
      "contiguous": false,
      "core_spec": null,
      "thread_spec": null,
      "cores_per_socket": null,
      "billable_tres": 2.000000,
      "cpus_per_task": null,
      "cpu_frequency_minimum": null,
      "cpu_frequency_maximum": null,
      "cpu_frequency_governor": null,
      "cpus_per_tres": "",
      "deadline": 0,
      "delay_boot": 0,
      "dependency": "",
      "derived_exit_code": 0,
      "eligible_time": 1647255451,
      "end_time": 1652439453,
      "excluded_nodes": "",
      "exit_code": 0,
      "features": "",
      "federation_origin": "slurm_testbed_1",
      "federation_siblings_active": "slurm_testbed_1",
      "federation_siblings_viable": "slurm_testbed_1,slurm_testbed_2",
      "gres_detail": [
      ],
      "group_id": 600,
      "job_id": 67109281,
      "job_resources": {
        "nodes": "slurm03",
        "allocated_cpus": 2,
        "allocated_hosts": 1,
        "allocated_nodes": {
          "0": {
            "sockets": {
              "0": "unassigned",
              "1": "unassigned"
            },
            "cores": {
              "0": "unassigned"
            },
            "memory": 8192,
            "cpus": 2
          }
        },
        "job_state": "RUNNING",
        "last_sched_evaluation": 1647255453,
        "licenses": "",
        "max_cpus": 0,
        "max_nodes": 0,
        "mcs_label": "",
        "memory_per_tres": "",
        "name": "submit_test",
        "nodes": "slurm03",
        "nice": null,
        "tasks_per_core": null,
        "tasks_per_node": 0,
        "tasks_per_socket": null,
        "tasks_per_board": 0,
        "cpus": 2,
        "node_count": 1,
      }
    }
  ]
}
```

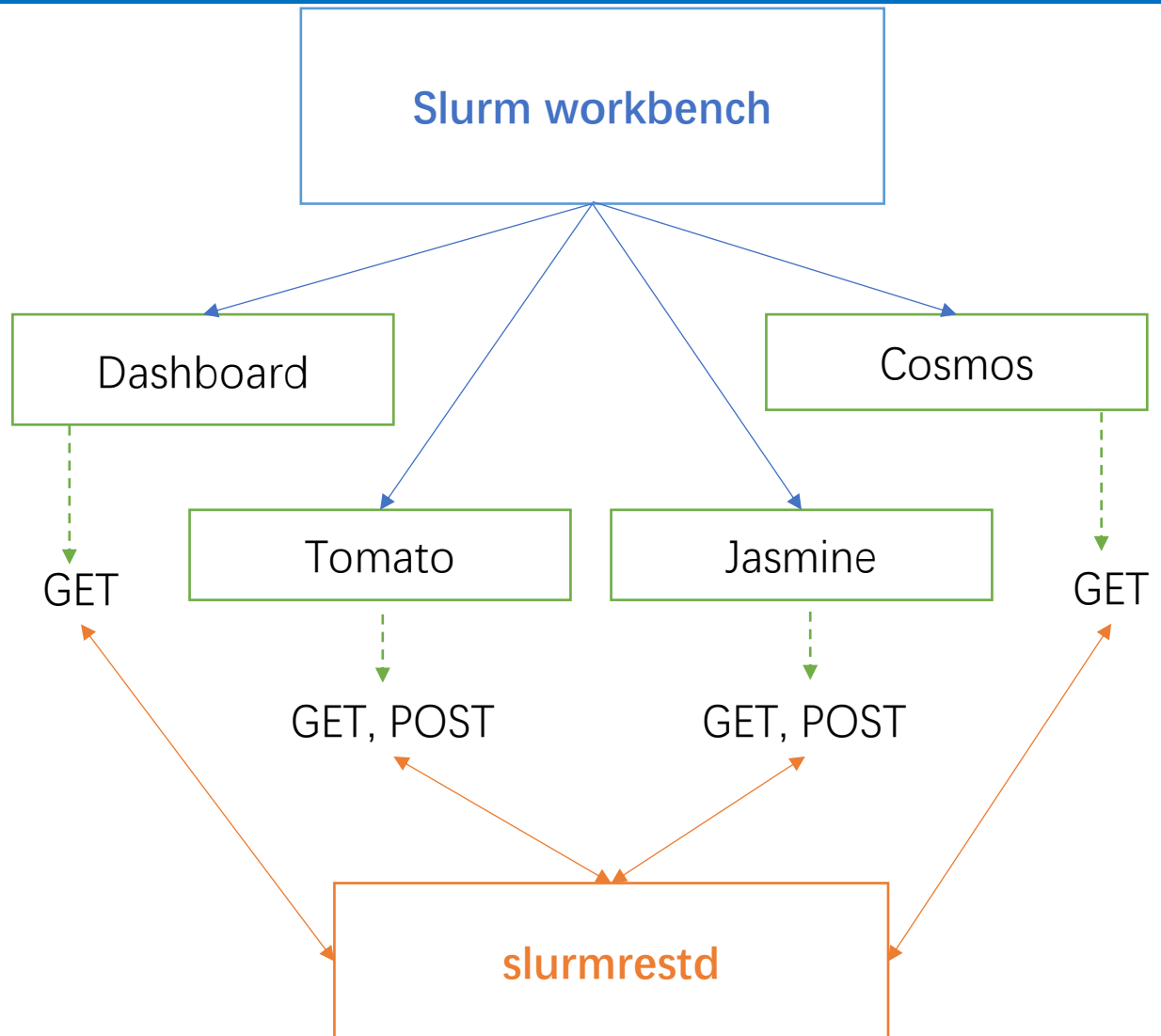


Development of the workbench



Architecture of the Slurm workbench

- Four sub-systems consisted:
 - Dashboard : cluster status display
 - Tomato : HPC-HTC jobs research
 - Jasmine : cluster job workload tool
 - Cosmos : accounting system
- Implementation:
 - Python Flask
 - MariaDB



Workbench - Dashboard

Overview of the Slurm Cluster

[Home](#) / Dashboard



196 CPU Nodes



8768
CPU cores



32 GPU Nodes



218
GPU cards



1160
CPU cores

Running CPU Jobs



87

27%

Running GCPU Jobs



43

13%

Running GPU Jobs



193

60%

11740 submitted jobs

11692 started jobs

11613 completed jobs

115 cancelled jobs

0 failed jobs

64 pending jobs

11 CPU partitions

7 GPU partitions

22 QOS

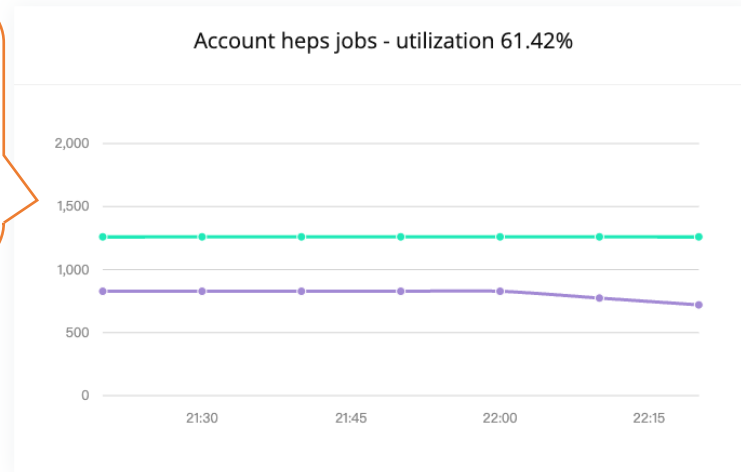
29 accounts

488 users



Workbench - Dashboard

Resource utilization of group heps in last hour.



Top 5 users ranked by consumed GPU hours in last day.

Top 5 User - GPU Resources				
1		huangzhenxiu group gpupwa	125.6259	Details
2		wangjun group junogpu	96.4764	Details
3		liujiayi16 group junogpu	92.0111	Details
4		marq group gpupwa	87.3390	Details
5		wangshi group mlgpu	62.5494	Details

Resources in each partition.

[Home](#) / [Dashboard](#) / [Nodes](#)

Partition - Node - Resource Table

Total resources of each partition **at present**

#	Partition	Total Nodes	Total CPU Cores	Total GPU Nodes	Memory per Node (GB)
1	heps	34	1224	0	180
2	cepcmpi	36	1696	0	110
3	biofasq	11	264	0	60
4	cac	7	336	0	90
5	raq	12	672	0	240
6	spub	20	1040	0	240
7	mbh	16	256	0	20
8	ali	16	576	0	500
9	ali	28	1792	0	240
10	gpu	24	864	190	360
11	lgpu	1	36	8	360
12	neuph	2	96	5	360



Workbench - Tomato

Submit Glidein Jobs

- A research system for HTC-HPC jobs.
- The key component is to submit Glidein jobs.
- After submission, Jobs from HTC cluster will be run in the Slurm cluster.
- Part of dHTC, see the next work presented by Xiaowei Jiang:

HTC Computing System

Job Parameters

Partition
spub

Account
u07

QOS
glidein

Memory per CPU
4096

Num. of CPU
56

Output File Name
/tmp/gliden_job_%j.out

Job Name
glidein

Job parameters

Job workload

```
echo "Glidenin job {SLURM_JOB_ID} starting"
date

# create a specific work directory for glidein job
SOFTWARE_DIR="/home/cc/jiangxw/tmp_j/hpc_glidein"
export GLID_WORK_DIR=/tmp/$SLURM_JOB_ID/
mkdir -p ${GLID_WORK_DIR}
cd ${GLID_WORK_DIR}

# start slot
${SOFTWARE_DIR}/startup.sh ihpc_hpc_spub.cfg

rm -rf $GLID_WORK_DIR

echo "Glidenin job {SLURM_JOB_ID} stopped."
date
```

Job workload

Generate

Generated Job Script

Job Script to Submit

```
#!/bin/bash

##### Part 1 : job parameters #####
#SBATCH --partition=spub
#SBATCH --account=u07
#SBATCH --qos=glidein
#SBATCH --job-name=glidein
#SBATCH --output=/tmp/gliden_job_%j.out
#SBATCH --ntasks-per-node=56
#SBATCH --nodes=1
#SBATCH --mem-per-cpu=4G

##### Part 2 : startd #####
echo "Glidenin job {SLURM_JOB_ID} starting"
date

# create a specific work directory for glidein job
SOFTWARE_DIR="/home/cc/jiangxw/tmp_j/hpc_glidein"
export GLID_WORK_DIR=/tmp/$SLURM_JOB_ID/
mkdir -p ${GLID_WORK_DIR}
cd ${GLID_WORK_DIR}

# start slot
${SOFTWARE_DIR}/startup.sh ihpc_hpc_spub.cfg

rm -rf $GLID_WORK_DIR

echo "Glidenin job {SLURM_JOB_ID} stopped."
date
```

Job script

Workbench - Tomato

Submit Jobs

3

Submit

Check Job Status

Number of jobs in a job array

Submitted jobs

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
67109296_1	spub	glidein	root	R	0:04	1	spub001
67109296_2	spub	glidein	root	R	0:04	1	spub002
67109296_3	spub	glidein	root	R	0:04	1	spub003

Check job status after submission



Workbench - Jasmine

- A job toolkit to test Slurm cluster.
- Generate and submit jobs base on parameters.
 - User
 - Time
 - Number of jobs
 - Parallel degree
 - Job size
- Application scenario:
 - Overall test during summer maintenance.
 - Research on Slurm scheduling algorithms.

Generate & Submit Jobs

CPU APP PARTITIONS

GPU APP PARTITIONS

Job Parameters

Scheduler

Slurm

Num. of users/label>

Num. of users

User

Time pattern

random

Runtime Duration

in seconds

Time

Interval of Submit

in seconds

Pattern of Parallel Degree

random

Seed of Parallel Degree

para seed

Parallel Degree

Pattern of Job Bunch

random

Seed of Job Bunch

job bunch seed

Number of Jobs

Pattern of Job Size

random

Seed of Job Size

job size seed

Factor of Job Size

job size factor

Job size

Job Workload

job workload

submit



Workbench - Cosmos

- Accounting system

Welcome to Slurm Job Query Interface

Please input at least one field to query jobs

Query Form

Query Results in a Table

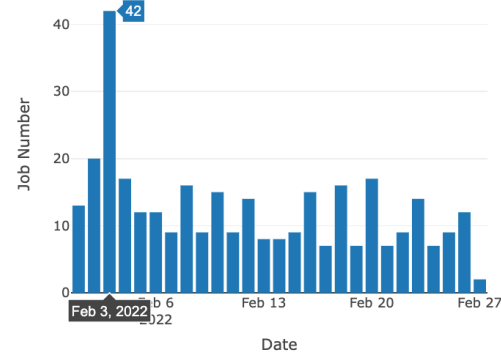
Query Result Tables

Summary					
	Records	Jobs	CPU * Hours	GPU * Hours	Users
0	53440	27065	15334.075833	2317.978889	6

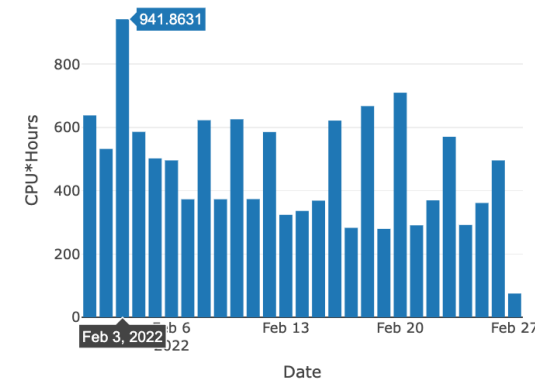
Job Details										
	job_id	user	group	hostname	res_type	count	time_submit	time_start	time_end	duration
0	1017710	zfliu	lqcd	gpu031	cpu	4	2022-02-01 18:27:12	2022-02-01 18:27:13	2022-02-02 05:28:48	39695
1	1017711	zfliu	lqcd	gpu032	cpu	4	2022-02-01 18:29:12	2022-02-01 18:29:12	2022-02-02 05:13:17	38645
2	1017712	zfliu	lqcd	gpu033	cpu	4	2022-02-01 18:30:19	2022-02-01 18:30:21	2022-02-02 05:19:39	38958
3	1017958	zfliu	lqcd	gpu022	cpu	8	2022-02-02 14:18:37	2022-02-02 14:18:39	2022-02-02 16:08:41	6602
4	1017958	zfliu	lqcd	gpu022	gpu	8	2022-02-02 14:18:37	2022-02-02 14:18:39	2022-02-02 16:08:41	6602
5	1017959	zfliu	lqcd	gpu023	cpu	8	2022-02-02 14:18:49	2022-02-02 14:18:50	2022-02-02 16:21:42	7372
6	1017959	zfliu	lqcd	gpu023	gpu	8	2022-02-02 14:18:49	2022-02-02 14:18:50	2022-02-02 16:21:42	7372
7	1017960	zfliu	lqcd	gpu026	cpu	8	2022-02-02 14:18:56	2022-02-02 14:18:56	2022-02-02 16:10:29	6693
8	1017960	zfliu	lqcd	gpu026	gpu	8	2022-02-02 14:18:56	2022-02-02 14:18:56	2022-02-02 16:10:29	6693
9	1017961	zfliu	lqcd	gpu027	cpu	8	2022-02-02 14:19:06	2022-02-02 14:19:06	2022-02-02 16:07:33	6507
10	1017961	zfliu	lqcd	gpu027	gpu	8	2022-02-02 14:19:06	2022-02-02 14:19:06	2022-02-02 16:07:33	6507
11	1017962	zfliu	lqcd	gpu028	cpu	8	2022-02-02 14:19:13	2022-02-02 14:19:14	2022-02-02 16:04:42	6328
12	1017962	zfliu	lqcd	gpu028	gpu	8	2022-02-02 14:19:13	2022-02-02 14:19:14	2022-02-02 16:04:42	6328
13	1017963	zfliu	lqcd	gpu029	cpu	8	2022-02-02 14:19:18	2022-02-02 14:19:19	2022-02-02 16:17:46	7107
14	1017963	zfliu	lqcd	gpu029	gpu	8	2022-02-02 14:19:18	2022-02-02 14:19:19	2022-02-02 16:17:46	7107
15	1017964	zfliu	lqcd	gpu031	cpu	8	2022-02-02 14:19:21	2022-02-02 14:19:22	2022-02-02 16:14:37	6915

Query Results in Charts

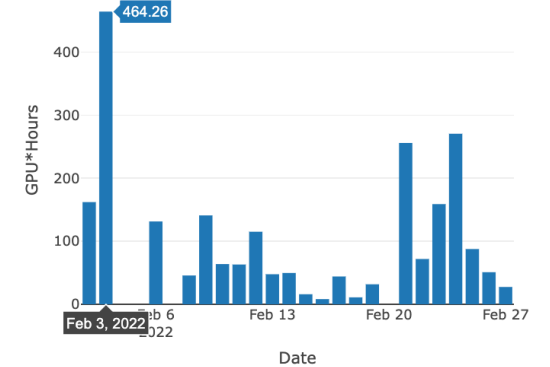
Barplot of CPU Job Number



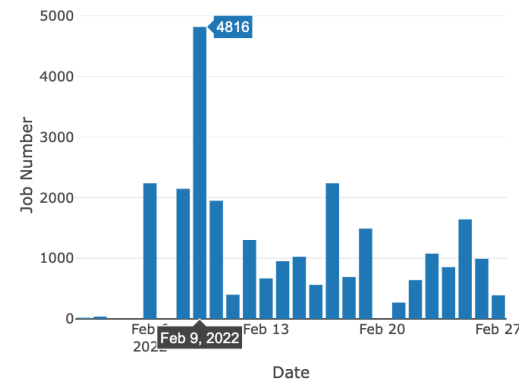
Barplot of CPU*Hours



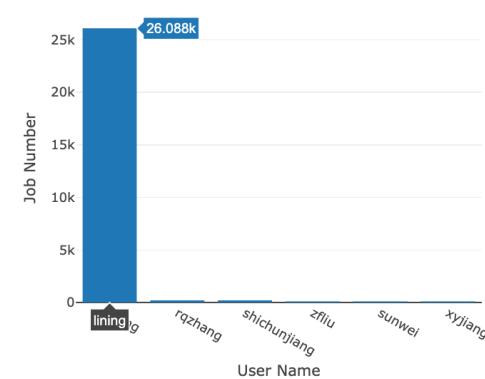
Barplot of GPU*Hours



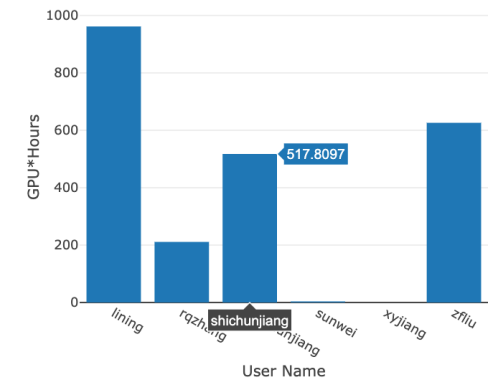
Barplot of GPU Job Number



Barplot of GPU Job Number of Users



Barplot of GPU*Hours of Users



Conclusion & Next



Conclusion

- RESTful Slurm workbench is developed because of maintenance and research requirements.
- Slurm workbench is consisted with four sub-systems.
- GET & POST methods of Slurm REST APIs are called by workbench sub-systems.
- Pre-production, only for administrators & researchers at present.



Next ...

- JWT tokens renew mechanism.
- Proxy + cache architecture for better performance.
- Save intermediate job descriptions of Jasmine in Database.
- Cosmos REST APIs open to other systems.



Some thoughts and experiences

- More POST/PUT methods are favorable to manage slurmdbd.
 - Management Board
- Additional user authentication is recommended.
 - More secure and flexible security policy
- Architecture of rest_auth/jwt + proxy is better.
 - To provide web services for common users.



Thanks & Questions

