基于REST API的Slurm Workbench原型系统开发

杜然 duran@ihep.ac.cn，高能所计算中心

第20届全国科学计算与信息化会议

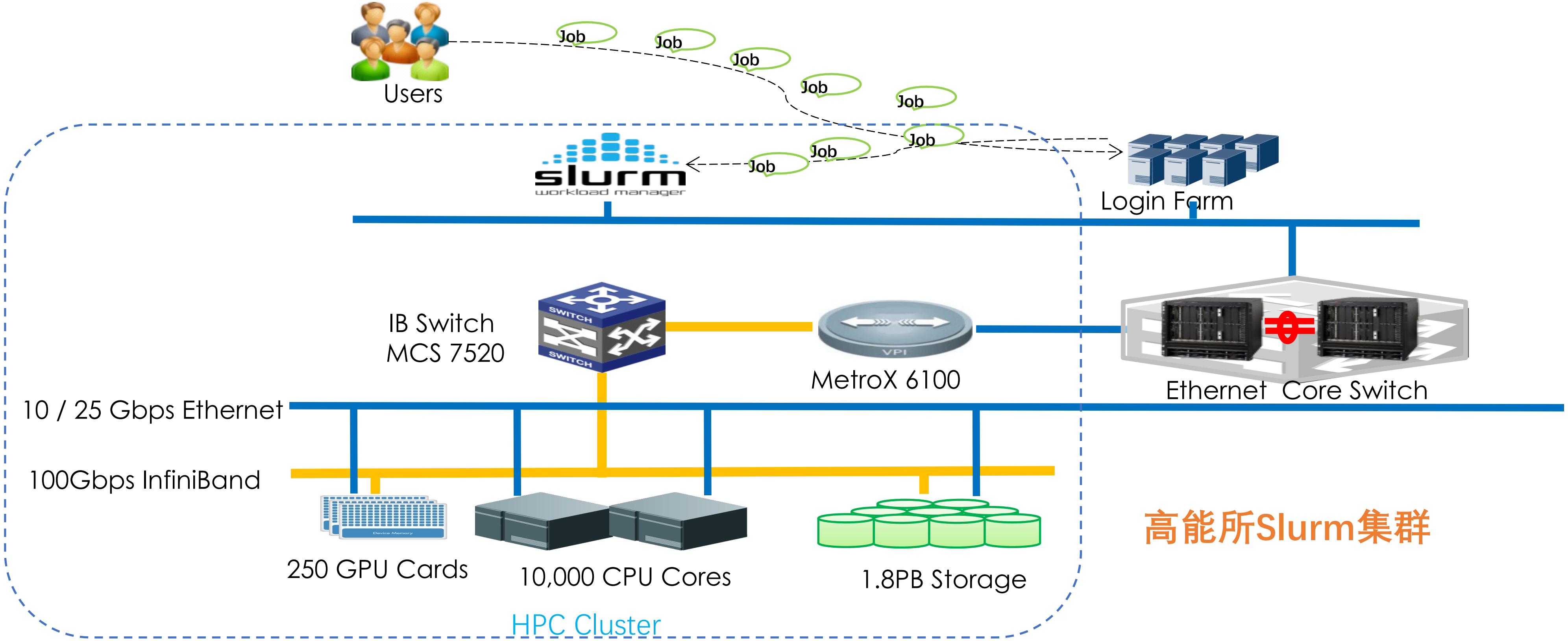
2023-07-10

Outline

- 开发背景及需求
- Slurm REST API
- Slurm Workbench原型系统
- 总结及后续开发计划

背景 - 高能所Slurm集群

- 支持CPU并行计算和GPU异构加速计算
- ～1万CPU核，～250张GPU卡，1.8PB Lustre FS
- 支持20个组、约350活跃用户（2017.02～今）



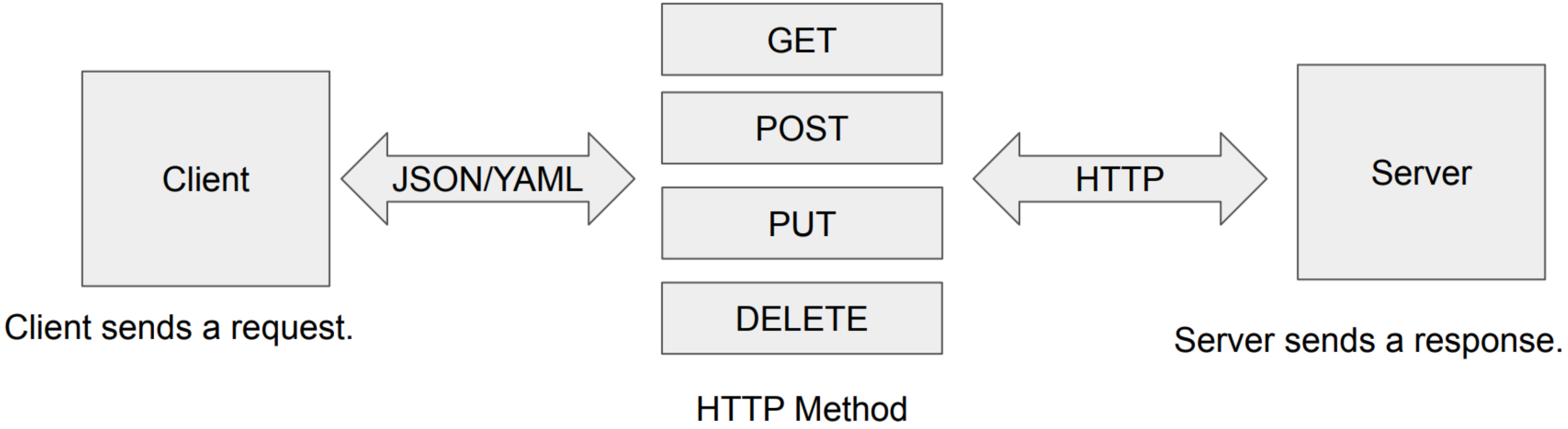
高能所Slurm集群

需求 – 支持研究和测试的应用系统

- Slurm集群研究和测试需求
  - CPU并行计算应用：HTC-HPC作业调度
  - GPU异构加速应用：静态QOS ceiling touch问题研究
  - Slurm调度插件性能测试
  - 。。。
- 支持完整的研究和测试流程
  - 作业负载生成、批量作业提交、集群状态查看、数据收集和分析
- Slurm自20.11版本起增加REST API
  - 为应用系统开发提供新的接口
  - 使用统一的技术栈整合已有的应用系统

REST简介

- REST：REpresentational State Transfer， Roy Fielding，2000
- 基于HTTP协议、使用URL作为资源的唯一标识
- CS多层级架构，具备无状态、可缓存的特点

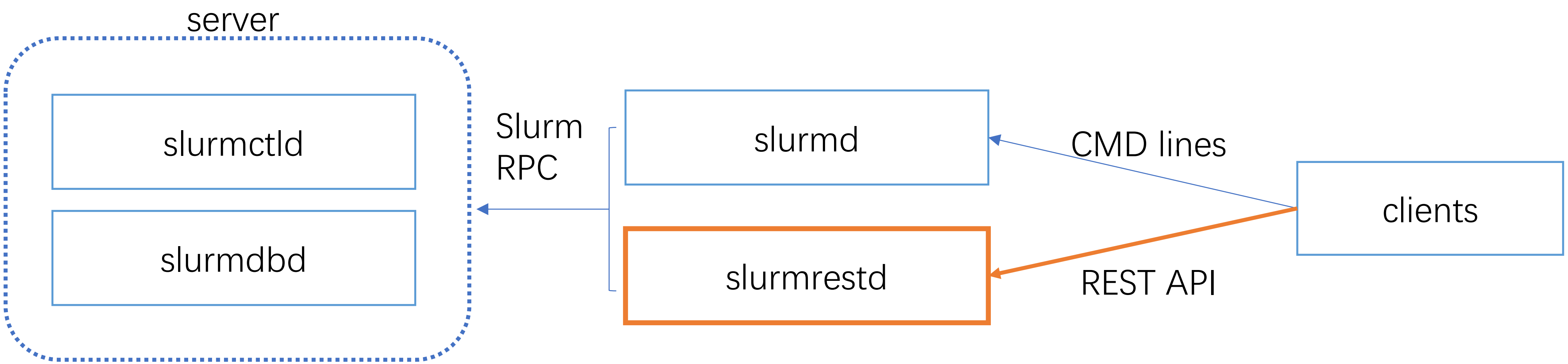


RESTful系统工作流

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

Slurm新架构

- 自20.11版本起，Slurm增加了slurmrestd daemon
- Slurmrestd位于server端（slurmctld/slurmdbd）与client端之间
  - 在此之前，client端可通过sbatch/sinfo等命令与server端交互
- 新架构符合REST结构和设计思想
  - 多层级架构：client端可以是应用系统

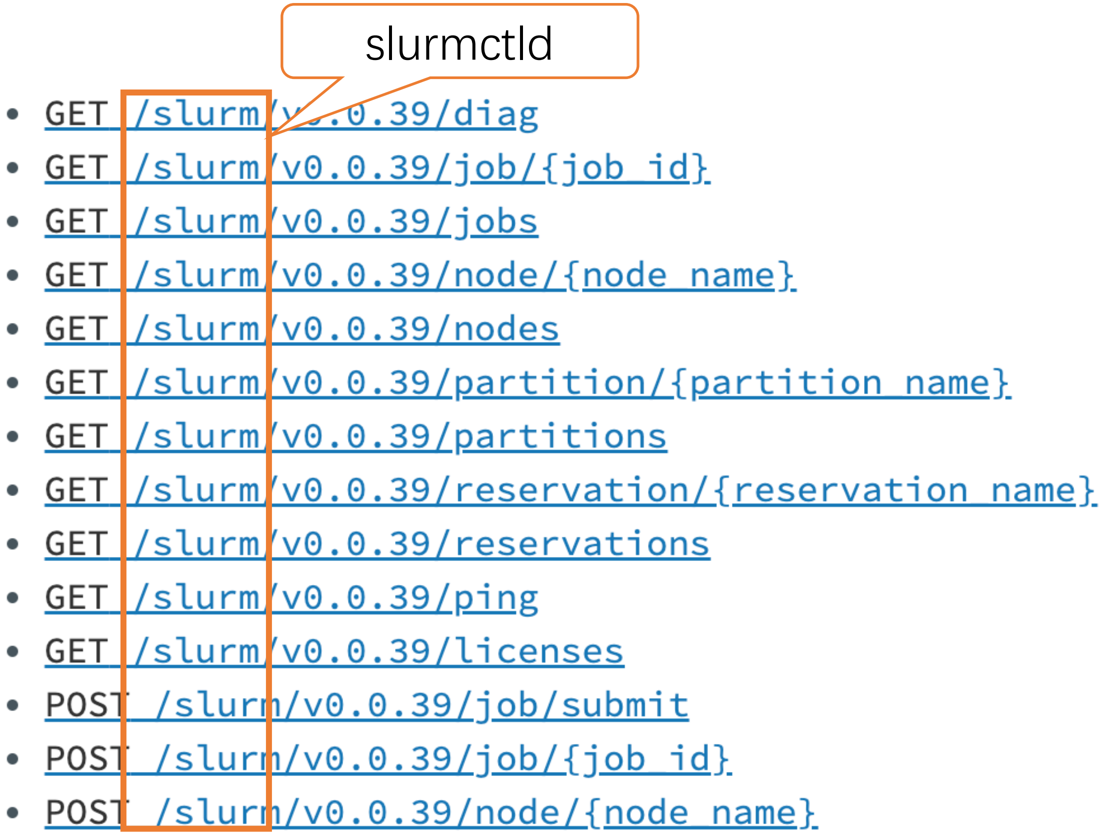


增加了slurmrestd的Slurm新架构

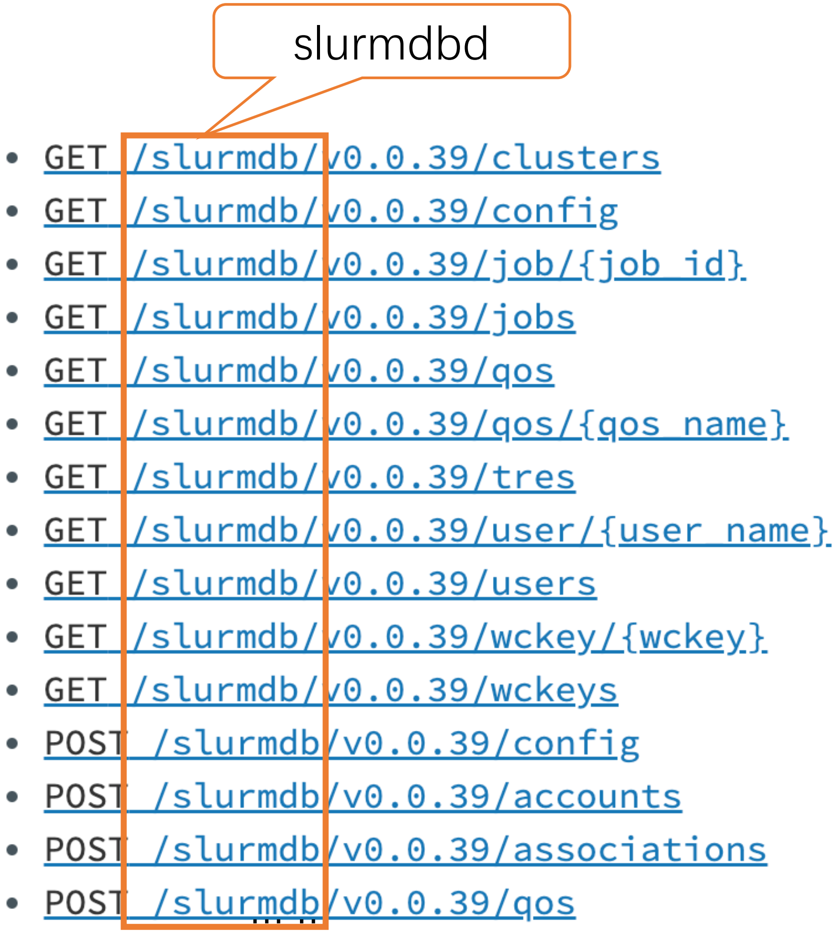
Reference : https://slurm.schedmd.com/SLUG20/REST_API.pdf

Slurm REST APIs

- Slurm REST API索引
- 提供GET, POST, DELETE方法， 无PUT方法
  - 20.11，21.08，22.05，23.02四个版本



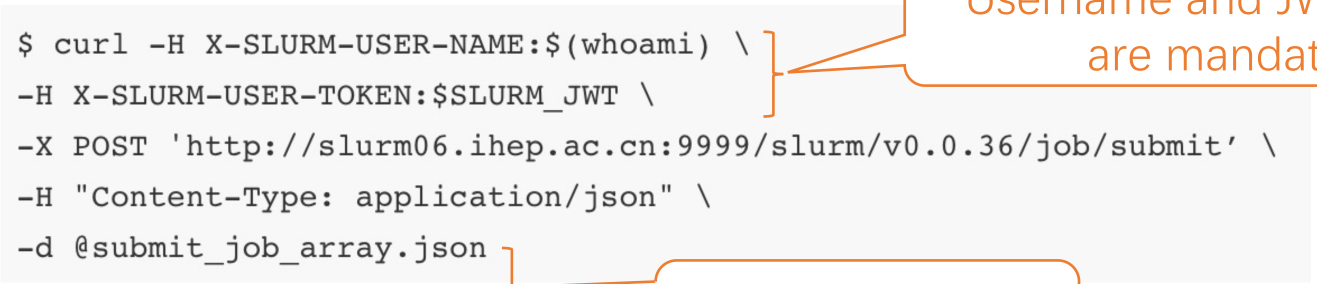
- GET /slurm/v0.0.39/diag
- GET /slurm/v0.0.39/job/{job_id}
- GET /slurm/v0.0.39/jobs
- GET /slurm/v0.0.39/node/{node_name}
- GET /slurm/v0.0.39/nodes
- GET /slurm/v0.0.39/partition/{partition_name}
- GET /slurm/v0.0.39/partitions
- GET /slurm/v0.0.39/reservation/{reservation_name}
- GET /slurm/v0.0.39/reservations
- GET /slurm/v0.0.39/ping
- GET /slurm/v0.0.39/licenses
- POST /slurm/v0.0.39/job/submit
- POST /slurm/v0.0.39/job/{job_id}
- POST /slurm/v0.0.39/node/{node_name}



- GET /slurmdb/v0.0.39/clusters
- GET /slurmdb/v0.0.39/config
- GET /slurmdb/v0.0.39/job/{job_id}
- GET /slurmdb/v0.0.39/jobs
- GET /slurmdb/v0.0.39/qos
- GET /slurmdb/v0.0.39/qos/{qos_name}
- GET /slurmdb/v0.0.39/tres
- GET /slurmdb/v0.0.39/user/{user_name}
- GET /slurmdb/v0.0.39/users
- GET /slurmdb/v0.0.39/wckey/{wckey}
- GET /slurmdb/v0.0.39/wckeys
- POST /slurmdb/v0.0.39/config
- POST /slurmdb/v0.0.39/accounts
- POST /slurmdb/v0.0.39/associations
- POST /slurmdb/v0.0.39/qos

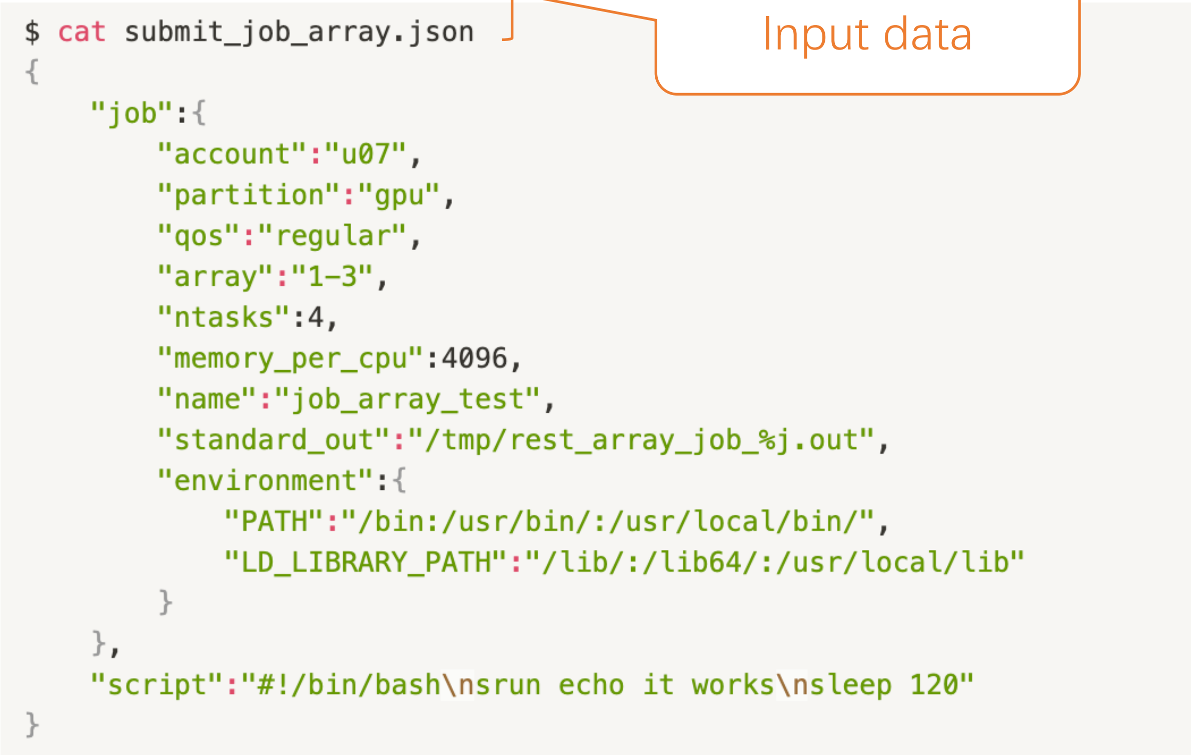
Slurm REST API - POST方法样例

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
```

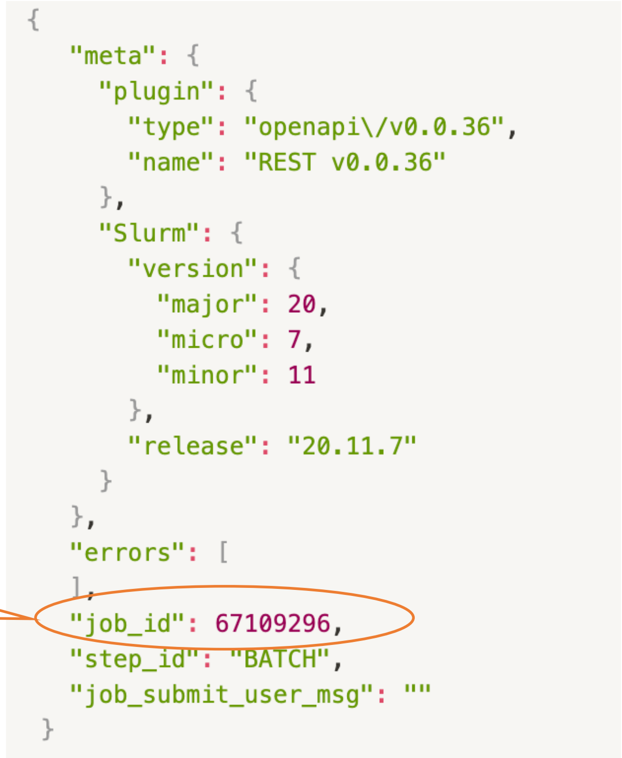
Username and JWT tokens are mandatory



```
$ 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\nsrun echo it works\nsleep 120"
}
```

Input data

Response

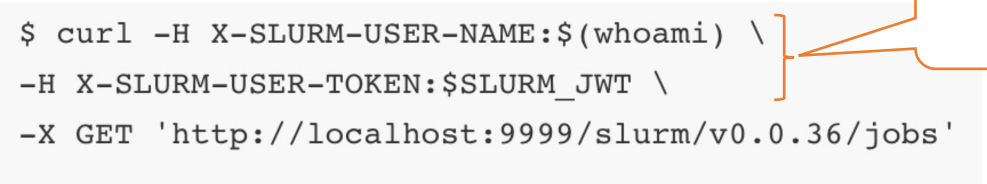


```
{
   "meta": {
     "plugin": {
       "type": "openapi\/v0.0.36",
       "name": "REST v0.0.36"
     },
     "Slurm": {
       "version": {
         "major": 20,
         "micro": 7,
         "minor": 11
       },
       "release": "20.11.7"
     }
   },
   "errors": [
   ],
   "job_id": 67109296,
   "step_id": "BATCH",
   "job_submit_user_msg": ""
 }
```

Return a job ID if successful.

Slurm REST API - GET方法样例

Request

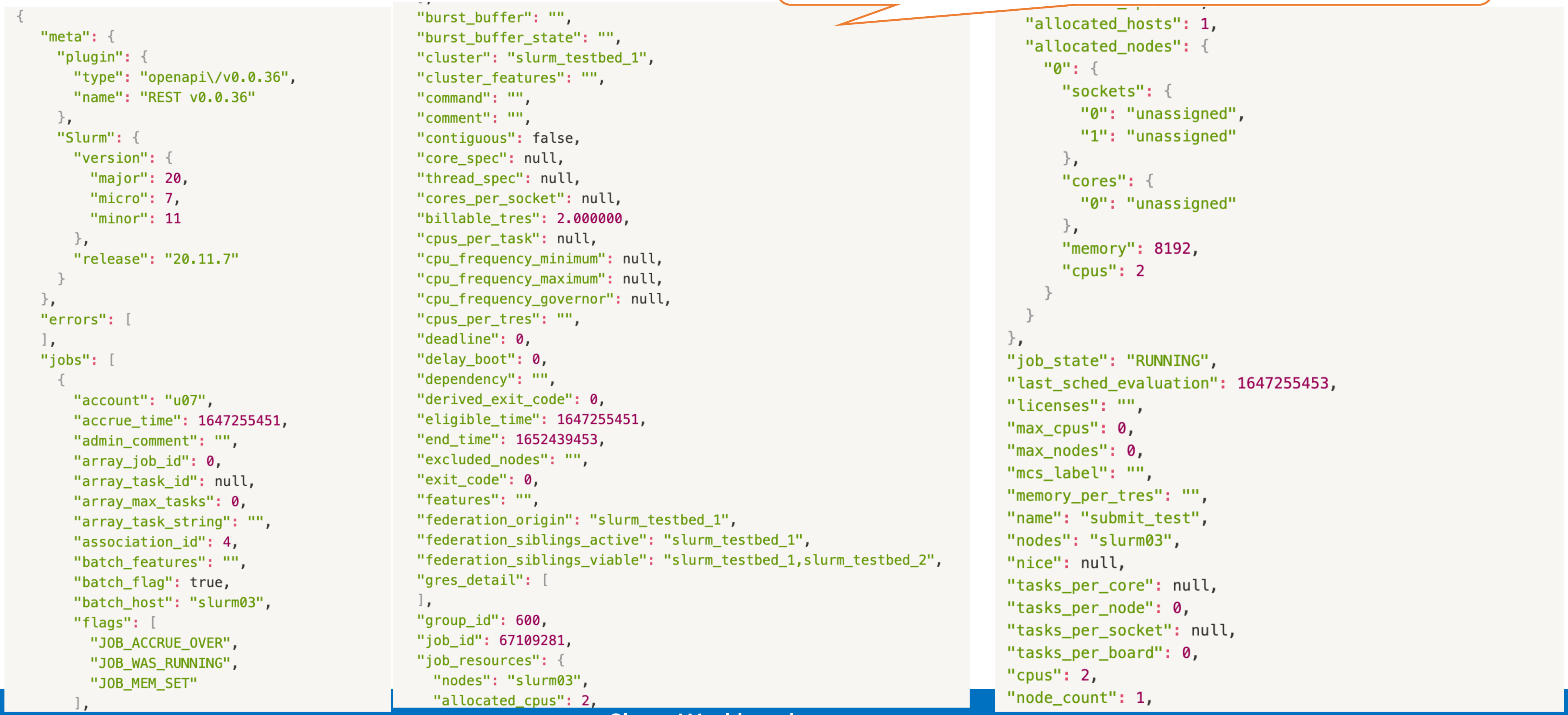


```
$ curl -H X-SLURM-USER-NAME:$(whoami) \
-H X-SLURM-USER-TOKEN:$SLURM_JWT \
-X GET 'http://localhost:9999/slurm/v0.0.36/jobs'
```

Username and JWT tokens are mandatory

Return detailed job description if successful

Response



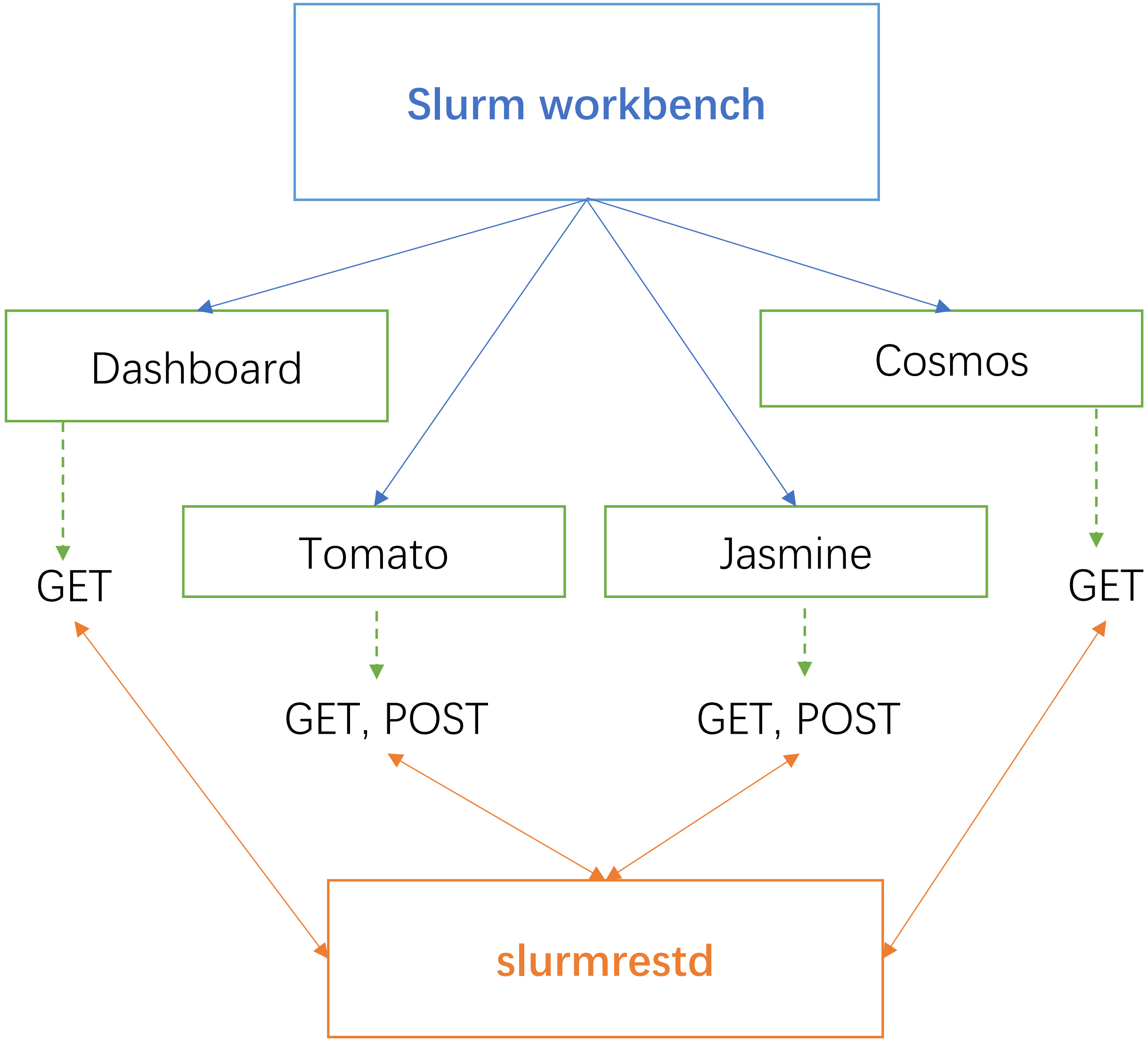
```
{
  "meta": {
    "plugin": {
      "type": "openapi\/v0.0.36",
      "name": "REST v0.0.36"
    },
    "Slurm": {
      "version": {
        "major": 20,
        "micro": 7,
        "minor": 11
      },
      "release": "20.11.7"
    }
  },
  "errors": [
  ],
  "jobs": [
    {
      "account": "u07",
      "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,
```

Slurm workbench

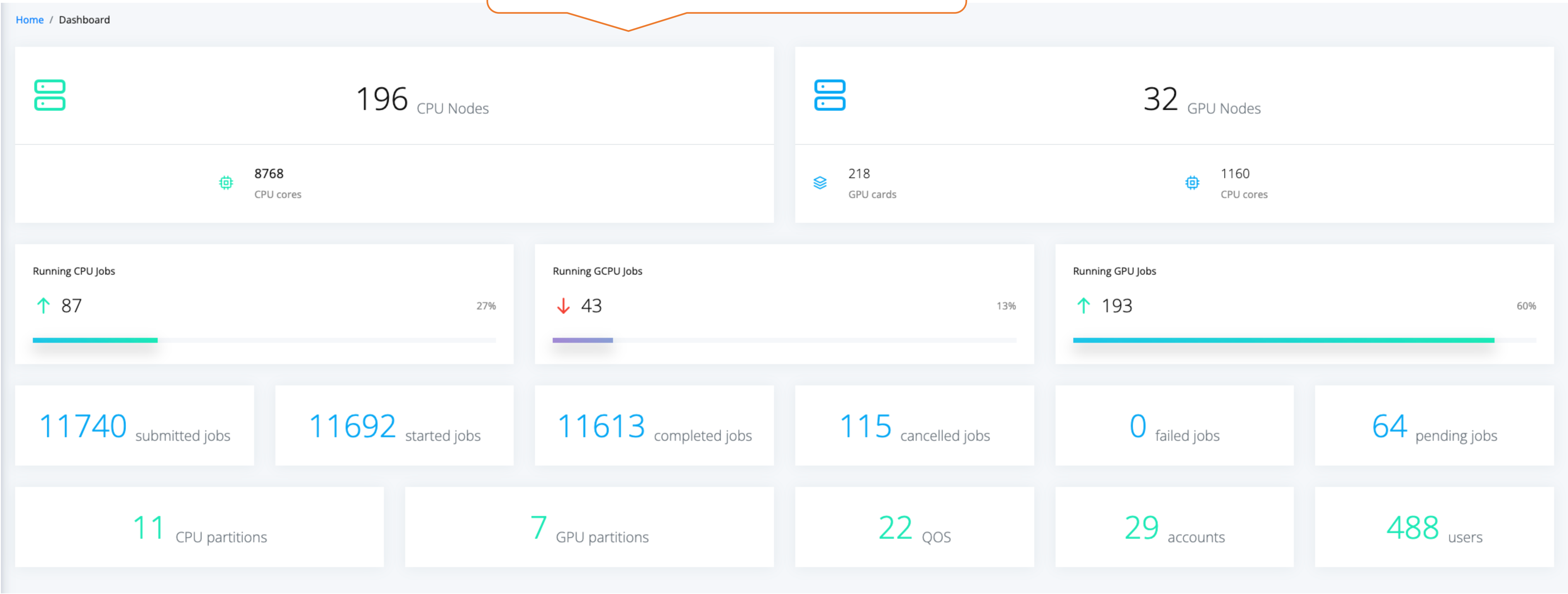
- Slurm workbench的子系统
  - Dashboard : 集群状态展板
  - Tomato : HPC-HTC作业研究
  - Jasmine : 集群作业负载生成及分析
  - Cosmos : 集群记账
  - 。。。
- 开发技术路线:
  - 前端：Python Flask
  - 后端：Python
  - 历史数据：MariaDB + Redis
  - 数据获取：Slurm REST API + 其他数据源分析



Slurm Workbench子系统

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

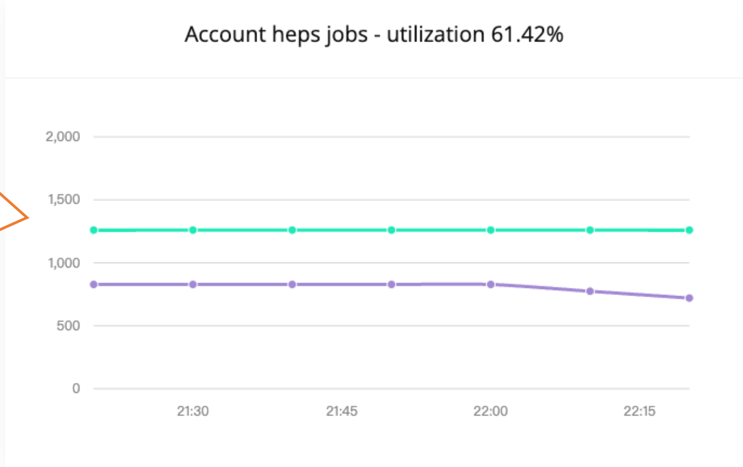
22 QOS

29 accounts

488 users

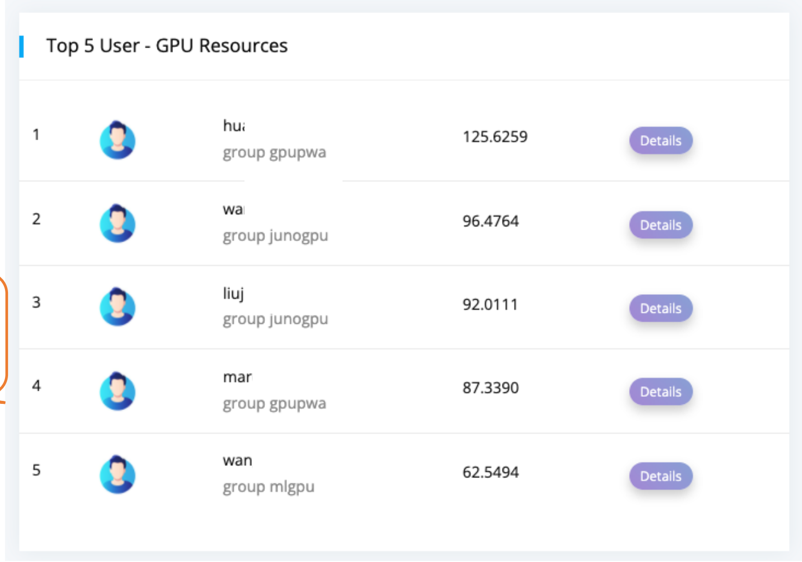
Workbench - Dashboard

Resource utilization



Account heps jobs - utilization 61.42%

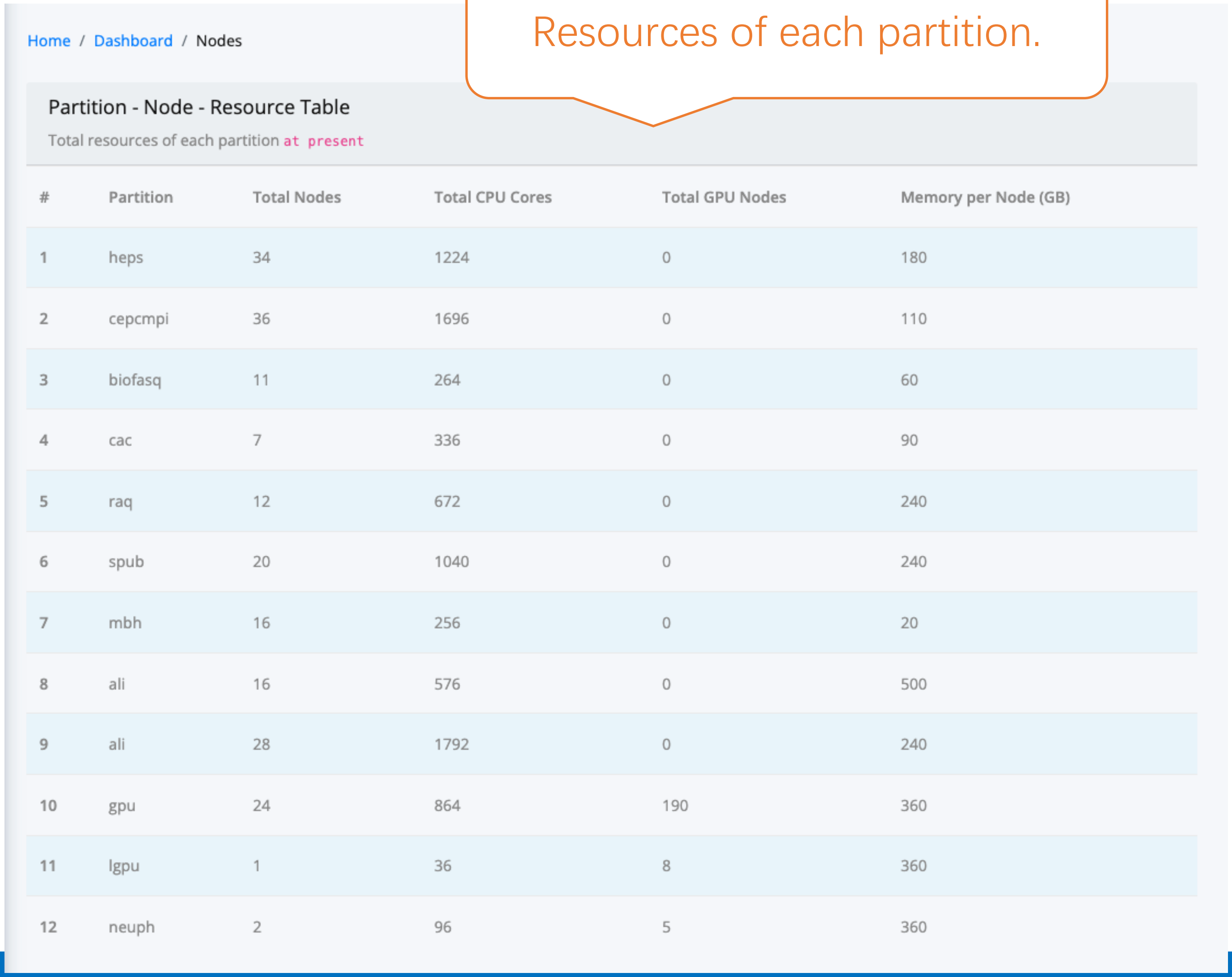
Top 5 users



Top 5 User - GPU Resources

1	hua group gpupwa	125.6259	Details
2	wa group junogpu	96.4764	Details
3	liuj group junogpu	92.0111	Details
4	mar group gpupwa	87.3390	Details
5	wan group mlgpu	62.5494	Details

Resources of each partition.



Home / Dashboard / Nodes

Partition - Node - Resource Table

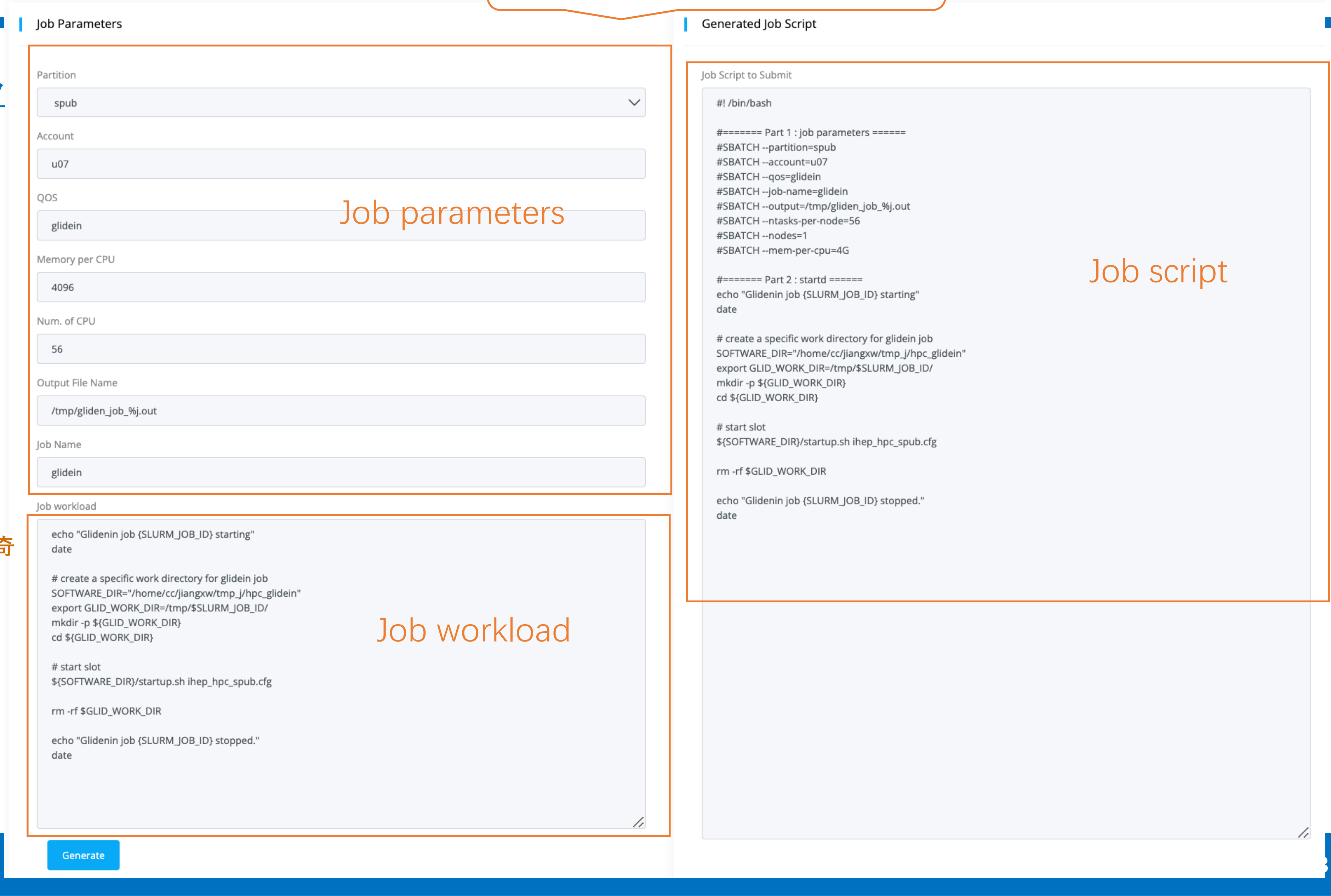
Total resources of each partition `at present`

#	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

- 研究HTC-HPC跨集群作业
- 关键功能是提交特殊的Glidein作业
- Glidein作业运行后，来自HTCondor集群的作业可运行在Slurm集群的计算节点中
- Glidein作业详细介绍：
  - 基于集群的分布式计算平台，郭超奇

Submit Glidein Jobs



Job Parameters

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 workload

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 ihep_hpc_spub.cfg

rm -rf $GLID_WORK_DIR

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

Generate

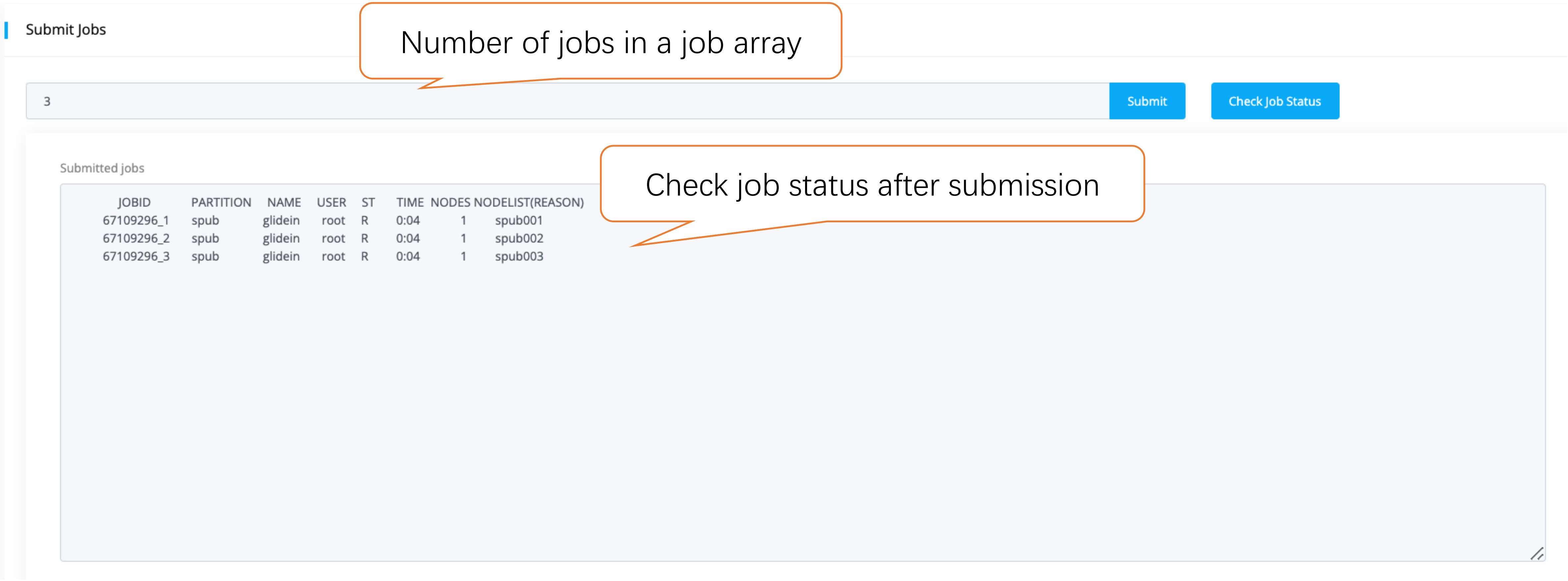
Generated Job Script

Job Script to Submit

Job script

```
#! /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 ihep_hpc_spub.cfg

rm -rf $GLID_WORK_DIR

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

Workbench - Tomato



Submit Jobs

3

Submitted jobs

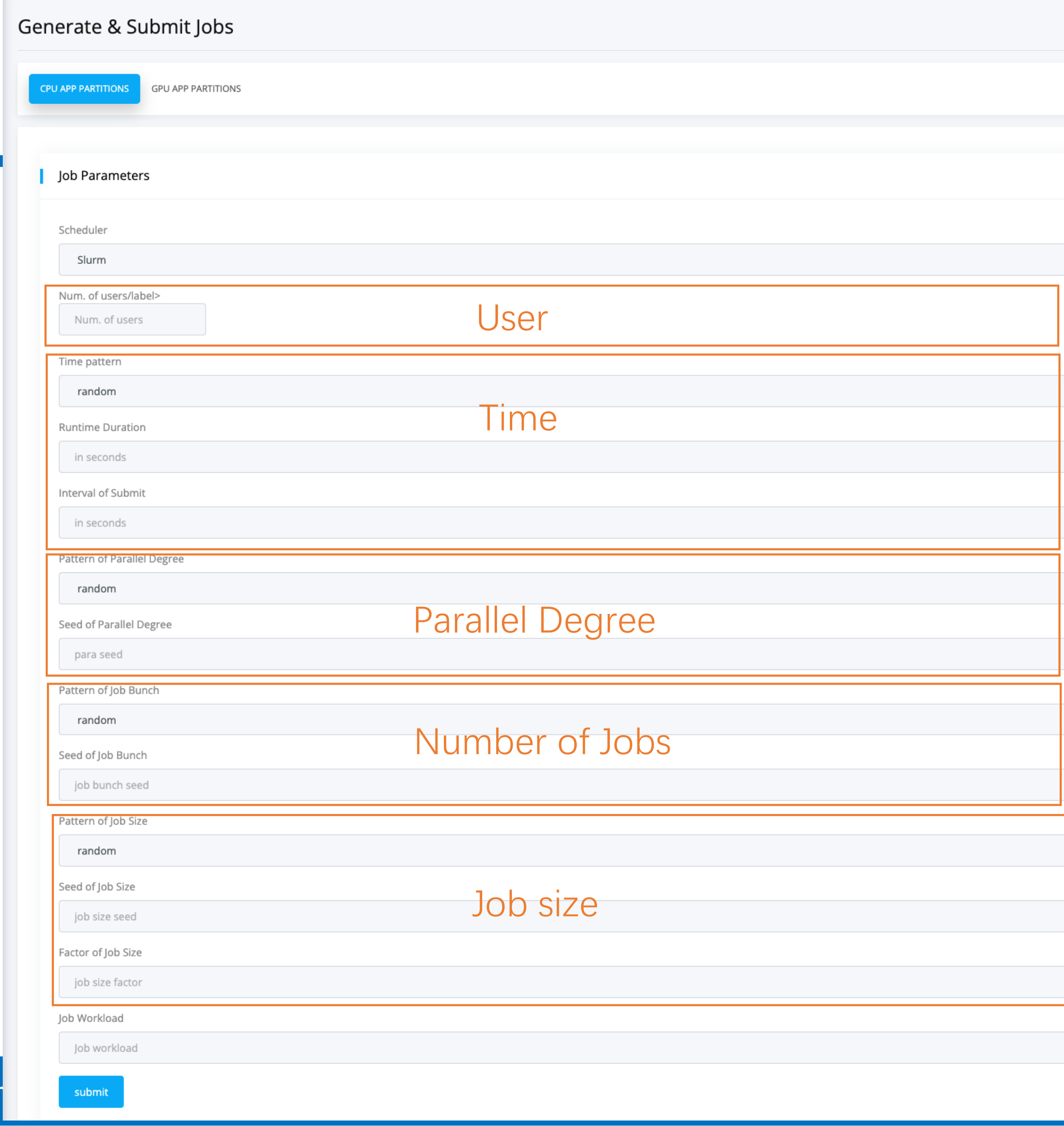
```
     JOBID      PARTITION   NAME    USER  ST    TIME  NODES NODELIST(REASON)
  67109296_1    spub       glidein  root   R    0:04      1     spub001
  67109296_2    spub       glidein  root   R    0:04      1     spub002
  67109296_3    spub       glidein  root   R    0:04      1     spub003
```

Number of jobs in a job array

Check job status after submission

Workbench - Jasmine

- 用于测试Slurm集群功能和性能
- 应用场景举例
  - 测试高温假升级后的全机作业运行
  - 研究Slurm作业调度插件的性能
- 可根据如下作业参数生成作业负载
  - User：用户
  - Time ：作业总运行时间
  - Number of jobs：作业数量
  - Parallel degree：作业并行度
  - Job size：单批量作业运行时间



Generate & Submit Jobs

Job Parameters

Scheduler: Slurm

Num. of users/label>: Num. of users

Time pattern: random

Runtime Duration: in seconds

Interval of Submit: in seconds

Pattern of Parallel Degree: random

Seed of Parallel Degree: para seed

Pattern of Job Bunch: random

Seed of Job Bunch: job bunch seed

Pattern of Job Size: random

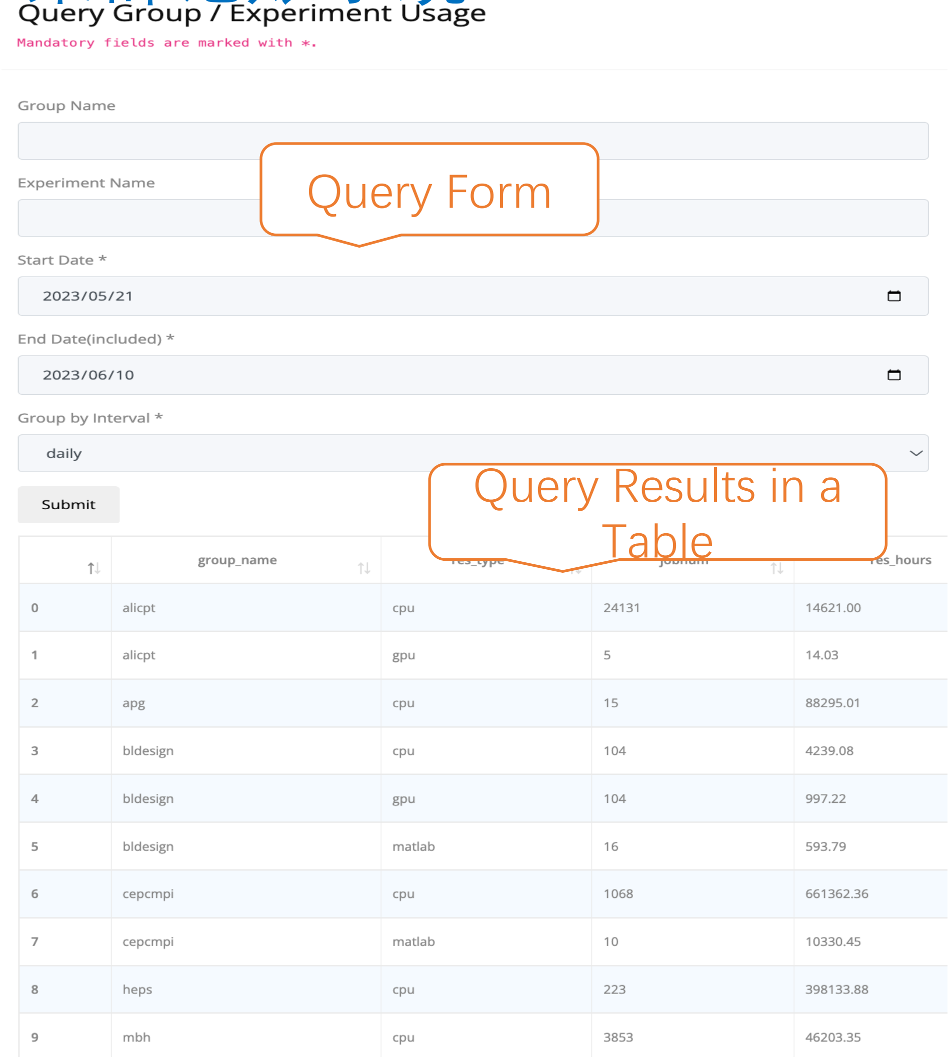
Seed of Job Size: job size seed

Factor of Job Size: job size factor

Job Workload: Job workload

Workbench - Cosmos

集群记账系统



Query Group / Experiment Usage

Mandatory fields are marked with *.

Group Name

Experiment Name

Start Date *

2023/05/21

End Date(included) *

2023/06/10

Group by Interval *

daily

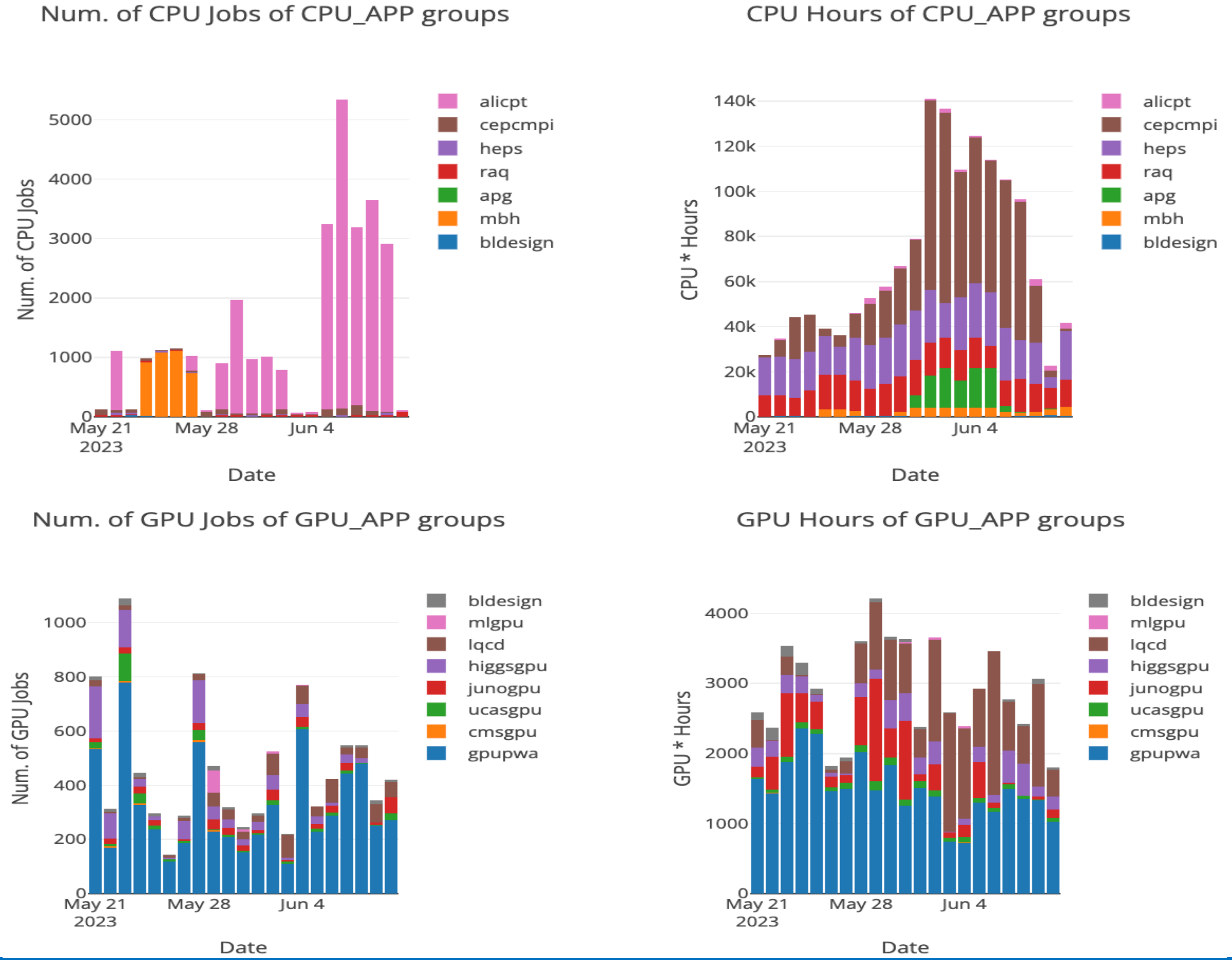
Submit

	group_name	res_type	jobnum	res_hours
0	alicpt	cpu	24131	14621.00
1	alicpt	gpu	5	14.03
2	apg	cpu	15	88295.01
3	bldesign	cpu	104	4239.08
4	bldesign	gpu	104	997.22
5	bldesign	matlab	16	593.79
6	cepcmpi	cpu	1068	661362.36
7	cepcmpi	matlab	10	10330.45
8	heps	cpu	223	398133.88
9	mbh	cpu	3853	46203.35

Query Form

Query Results in a Table

Query Results in Charts



Num. of CPU Jobs of CPU_APP groups

CPU Hours of CPU_APP groups

Num. of GPU Jobs of GPU_APP groups

GPU Hours of GPU_APP groups

总结 & 后续开发计划

- Slurm workbench原型系统
  - 为支持Slurm集群的研究测试
  - 基于Slurm REST API开发
  - 目前由四个子系统组成
- 已完成基本功能开发，后续开发计划：
  - 增加JWT tokens定期更新机制：与Kerberos tokens结合
  - 增加消息队列：支持jasmine系统大批量作业提交操作
  - 增加新REST API：支持对slurmdbd数据库的QOS修改
  - 与dHTC factory对接：管理glidein作业提交
  - 。。。

Thanks & Questions