The ability to fix a problem is as good as the tools and knowledge to fix it.
Outline

- How can I get all the information?
  - Commands and Logs
- How to be sure that the ball/problem is actually in Slurm’s court?
  - Setup checklist and best practices
- Why won’t my job run?
  - Backfill
- How to get (the best) support?
  - Bugzilla
Commands

- **Basic Info**
  - sinfo / squeue
  - sprio / sshare
  - smap

- **Config**
  - scontrol show config
  - sacctmgr show config

- **Details**
  - scontrol show [ partition | reservation... ]
  - sacctmgr show [ assoc | qos | event... ]

- **Stats & Diagnostic**
  - sdiag
  - sacctmgr show stats
  - sacctmgr show problem

- **Automation & Notifications**
  - strigger
  - HealthCheckProgram
  - UnkillableStepProgram

- **Autocorrection**
  - sacctmgr show RunawayJobs

- **External Process**
  - scontrol [listpids <job> | pidinfo <pid>]
  - taskset, stress...

- **Implicit Configs.**
  - `-o,--format → %letter VS -O,--Format → word`
  - Check running config, not only config files. Implicit Configs.
Details - `sacctmgr show [ assoc | qos | ... ]`

- Information about Limits or QoS of Users and Accounts (ie, of Associations)

```
$ sacctmgr show assoc tree Format=Account,User,Share,MaxTRES,QOS
Account  User  Share  MaxTRES  QOS
------  ----  ----  -------  ----
root    1     1     1      normal
root    root  1     1      normal
development  agil  70    1      normal,high
development  bob  70    1      normal,high
development  joe  70    1      normal,high
development  sue  70    1      normal,high
external  1     1     1      normal
external  jim  30    1      normal
external  joe  30    1      normal
```

```
$ sacctmgr show qos Format=Name,Priority,UsageFactor,MaxTRES,MaxTRESPU
Name   Priority UsageFactor       MaxTRES            MaxTRESPU
------  ------- ----------- ------------- -------------------
normal  5    1.000000                 cpu=80,gres/gpu=10
high   10    1.000000               cpu=800,gres/gpu=100
low    1     1.000000                   cpu=8,gres/gpu=1
```

Think more in Associations and Hierarchies and less in Users and Accounts

Do you really need `withassoc`?

Double check with `scontrol show assoc_mgr`

Copyright 2019 SchedMD

www.schedmd.com
Stats & Diagnostic - sdiag

- How many
  - Jobs being examined
  - Jobs being started by the backfill and quick scheduler
  - RPC calls are being executed
- How long take to process
  - the scheduling operations
  - RPC calls
- Who is executing RPC calls

$ sdiag

Server thread count: 3
Agent queue size: 0
Agent count: 0
DBD Agent queue size: 0

Jobs submitted: 523
Jobs started: 523
Jobs completed: 501
Jobs canceled: 3
Jobs failed: 19

Main schedule statistics (microsectons):
  Last cycle: 16
  Max cycle: 53
  Total cycles: 59
  Mean cycle: 20
  Mean depth cycle: 11
  Cycles per minute: 1
  Last queue length: 0

Backfilling stats
  Total backfilled jobs (since last slurm start): 0
  Total backfilled jobs (since last stats cycle start): 0
  Total cycles: 28
  Last cycle when: Wed Dec 30 15:33:18 2018
  Last cycle: 93
  Max cycle: 5433
  Last depth cycle: 0
  Last depth cycle (try sched): 0
  Last queue length: 0

Latency for 1000 calls to gettimeofday(): 15 microseconds

Remote Procedure Call statistics by message type
  REQUEST_RESOURCE_ALLOCATION (4001) count: 5 ave_time: 1880
  total_time: 94042
  REQUEST_JOB_READY (4019) count: 5 ave_time: 490
  total_time: 24520

Remote Procedure Call statistics by user
  student (1002) count: 32 ave_time: 1405
  total_time: 44973
  root (0) count: 0 ave_time: 0 total_time: 0

Pending RPC statistics
  No pending RPCs

Check size of queues and counts

Copyright 2019 SchedMD
www.schedmd.com
Stats & Diagnostic - sacctmgr show stats

- Provides detailed information about:
  - Rollups
  - How many Slurm DB RPC calls are being executed and by whom

```bash
$ sacctmgr show stats
Rollup statistics
Hour       count:746    ave_time:307765 max_time:223576198    total_time:229593291
Day        count:31     ave_time:2429   max_time:10972        total_time:75328
Month      count:1      ave_time:32007 max_time:32007        total_time:32007

Remote Procedure Call statistics by message type
DBD_CLUSTER_TRES         ( 1407) count:8948   ave_time:57837  total_time:517531274
DBD_JOB_COMPLETE         ( 1424) count:5      ave_time:17972  total_time:89864
DBD_FINI                 ( 1401) count:5      ave_time:4617   total_time:13852
SLURM_PERSIST_INIT       ( 6500) count:4      ave_time:341    total_time:1367
DBD_STEP_START           ( 1442) count:3      ave_time:256    total_time:1284
DBD_SEND_MULT_MSG        ( 1474) count:3      ave_time:1579   total_time:4738
DBD_STEP_COMPLETE        ( 1441) count:3      ave_time:3527   total_time:10581
DBD_SEND_MULTI_JOB_START ( 1472) count:3      ave_time:3527   total_time:10581
DBD_JOB_START            ( 1425) count:2      ave_time:1146   total_time:2292
DBD_NODE_STATE           ( 1432) count:2      ave_time:2427   total_time:4854
DBD_GET_USERS            ( 1415) count:1      ave_time:510    total_time:510
DBD_GET_ASSOCS           ( 1410) count:1      ave_time:1768   total_time:1768
DBD_GET_RES             ( 1478) count:1      ave_time:274    total_time:274
DBD_REGISTER_CTLD        ( 1434) count:1      ave_time:1065   total_time:1065
DBD_GET_TRES             ( 1486) count:1      ave_time:357    total_time:357
DBD_GET_FEDERATIONS      ( 1494) count:1      ave_time:477    total_time:477
DBD_GET_QOS              ( 1448) count:1      ave_time:208    total_time:208
DBD_GET_STATS            ( 1489) count:1      ave_time:387    total_time:387
DBD_GET_CONFIG           ( 1466) count:1      ave_time:99     total_time:99

Remote Procedure Call statistics by user
agil                (      1000) count:8987   ave_time:57601 total_time:517669088
```
Stats & Diagnostic - sacctmgr show problem

$ sacctmgr show problem

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Account</th>
<th>User</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>jim</td>
<td>User does not have a uid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>joe</td>
<td>User does not have a uid</td>
</tr>
</tbody>
</table>

Quick Poll?
Automation - strigger

- strigger manage event triggers, programs to be run on a variety of events
- Mostly used to notify sysadmins about failures, and quick containment

```
> strigger --set --primary_slurmctld_failure \  
   --program=/usr/sbin/primary_slurmctld_failure.bash

> cat /usr/sbin/primary_slurmctld_failure.bash
#!/bin/bash
# Resubmit trigger
strigger --set --primary_slurmctld_failure \  
   --program=/usr/sbin/primary_slurmctld_failure.bash
# Send notification
/bin/mail slurm_admin@site.com -s Primary_slurmctld_failure
```
Automation - strigger events

### Events

- primary_slurmctld_failure
- primary_slurmctld_resummed_control
- primary_slurmctld_acct_buffer_full
- backup_slurmctld_failure
- backup_slurmctld_assumed_control
- primary_slurmdbd_failure
- primary_slurmdbd_resummed_operation
- primary_database_failure
- primary_database_resummed_operation

### State Changes

- **Down** (compute node enters DOWN state)
- **Drained** (compute node enters DRAINED state)
- **Fail** (compute node enters FAILING state)

---

Not a complete list. See manpage of `strigger` for complete listing.

---

Underused? Quick Poll?
Automation - strigger events (for users)

- This mechanism can also be used to notify user of job-specific events
  - Node failure in a job's allocation
  - Job nearing time limit

```
> strigger --set --time --offset=600 --jobid=123 \
   --program=/home/student/time_msg.bash

> cat /home/student/time_msg.bash
!/bin/bash
# Send notification when job within 600 seconds of time limit
/bin/mail student@site.com -s "Check job time limit"
```
Automation - HealthChecks

Because a healthy node is more than just a healthy slurmd:

- **HealthCheckProgram**
  - Fully qualified pathname of a script to execute as user root periodically on all compute nodes that are not in the NOT_RESPONDING state.

- **HealthCheckInterval**
  - The interval in seconds between executions of HealthCheckProgram.

- **HealthCheckNodeState**
  - Identify what node states should execute the HealthCheckProgram.

Integration with [LBNL Node Health Check](#)
Well, Slurm also needs to be able to kill jobs/processes:

- **UnkillableStepProgram**
  - If the processes in a job step are determined to be unkillable for a period of time specified by the `UnkillableStepTimeout` variable, the program specified by `UnkillableStepProgram` will be executed.

- **UnkillableStepTimeout**
  - The length of time, in seconds, that Slurm will wait before deciding that processes in a job step are unkillable (after they have been signaled with SIGKILL) and execute `UnkillableStepProgram`.
$ sacctmgr show runaway
NOTE: Runaway jobs are jobs that don’t exist in the controller but have a start time and no end time in the database

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Partition</th>
<th>Cluster</th>
<th>State</th>
<th>TimeSubmit</th>
<th>TimeStart</th>
<th>TimeEnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>17614</td>
<td>allocation</td>
<td></td>
<td>cluster</td>
<td>PENDING 2019-08-15T11:15:11</td>
<td>Unknown</td>
<td>Unknown</td>
<td></td>
</tr>
</tbody>
</table>

Would you like to fix these runaway jobs?
(This will set the end time for each job to the latest out of the start, eligible, or submit times, and set the state to completed. Once corrected, this will trigger the rollup to reroll usage from before the earliest submit time of all the runaway jobs.)

(You have 30 seconds to decide)
(N/y): y

Runway Jobs mess with your accounting. Stay clean.

There are other autocorrection commands, but you shouldn't use them unless proper support advise you to do it.
$ sbatch -n 4 -w c6 --wrap "srun sleep 300"
Submitted batch job 395152
$ ssh c6
$ scontrol listpids 395152

<table>
<thead>
<tr>
<th>PID</th>
<th>JOBID</th>
<th>STEPID</th>
<th>LOCALID</th>
<th>GLOBALID</th>
</tr>
</thead>
<tbody>
<tr>
<td>21462</td>
<td>395152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21463</td>
<td>395152</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21464</td>
<td>395152</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>21465</td>
<td>395152</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>21446</td>
<td>395152</td>
<td>batch</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21448</td>
<td>395152</td>
<td>batch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>21449</td>
<td>395152</td>
<td>batch</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

$ scontrol pidinfo 21464
Slurm job id 395152 ends at Fri Sep 04 15:50:47 2020
slurm_get_rem_time is 31535906

NAME
scontrol - Used view and modify Slurm configuration and state.
SYNOPSIS
scontrol [OPTIONS...] [COMMAND...]
OPTIONS
pidinfo proc_id
Print the Slurm job id and scheduled termination time corre-
sponding to the supplied process id, proc_id, on the current
node. This will work only with processes on node on which scon-
trol is run, and only for those processes spawned by Slurm and
their descendants.
listpids [job_id[.step_id]] [NodeName]
Print a listing of the process IDs in a job step (if
JOBID.STEPID is provided), or all of the job steps in a job (if
job_id is provided), or all of the job steps in all of the jobs
on the local node (if job_id is not provided or job_id is "+").
This will work only with processes on the node on which scontrol
is run, and only for those processes spawned by Slurm and their
descendants. Note that some Slurm configurations (ProctrackType
value of pgid) are unable to identify all processes associated
with a job or job step.

Protrack Issues
(eg CG)
use to be related to
pstree / cgroups
External Process - taskset

NAME

taskset - set or retrieve a process's CPU affinity

SYNOPSIS

taskset [options] mask command [argument...]

taskset [options] -p [mask] pid

OPTIONS

-c, --cpu-list

Interpret mask as numerical list of processors instead of a bitmask. Numbers are separated by commas and may include ranges. For example: 0,5,8-11.

-p, --pid

Operate on an existing PID and do not launch a new task.

Small “whereami” apps help a lot troubleshooting bindings, MPI, GPU ID...
External Process - stress(-ng)

NAME
stress-ng - a tool to load and stress a computer system

SYNOPSIS
stress-ng [OPTION [ARG]] ...

DESCRIPTION
stress-ng will stress test a computer system in various selectable ways. It was designed to exercise various physical subsystems of a computer as well as the various operating system kernel interfaces. stress-ng also has a wide range of CPU specific stress tests that exercise floating point, integer, bit manipulation and control flow.

Stressing jobs to stress clusters is stressing...
but stress tests save lives!

Prevent is better than cure
Ensure config works as expected or Replicate an issue
Logs are not an easy thing to deal with, but are (and will be) an important tool.
Logs - Two ways to increase logs

● Modular
  ○ Specific subsystems
  ○ DebugFlags (slurmctld and slurmdbd)

Be more modular than verbose

● Levels
  ○ More/Less verbosity
  ○ slurmctld
    ■ Slurmctld(Syslog)Debug
    ■ SlurmSchedLogLevel
  ○ slurmdbd
    ■ DebugLevel(Syslog)
  ○ slurmd
    ■ Slurmd(Syslog)Debug
## Logs - List of Flags and Levels

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backfill</td>
<td>Backfill scheduling</td>
</tr>
<tr>
<td>BurstBuffer</td>
<td>BurstBuffer state</td>
</tr>
<tr>
<td>CPU_Bind</td>
<td>Binding of tasks to CPUs</td>
</tr>
<tr>
<td>CpuFrequency</td>
<td>CPU governor and frequency management</td>
</tr>
<tr>
<td>Energy</td>
<td>Energy use</td>
</tr>
<tr>
<td>ExtSensors</td>
<td>External sensors</td>
</tr>
<tr>
<td>Gres</td>
<td>Generic Resource allocation and use</td>
</tr>
<tr>
<td>Gang</td>
<td>Gang scheduling</td>
</tr>
<tr>
<td>License</td>
<td>License scheduling</td>
</tr>
<tr>
<td>Priority</td>
<td>Job priority</td>
</tr>
<tr>
<td>Power</td>
<td>System power management</td>
</tr>
<tr>
<td>Reservations</td>
<td>Advanced reservations</td>
</tr>
<tr>
<td>SelectType</td>
<td>Resource selection</td>
</tr>
<tr>
<td>Steps</td>
<td>Job step activities</td>
</tr>
<tr>
<td>Switch</td>
<td>Network resources (e.g. the switch plugin)</td>
</tr>
<tr>
<td>Triggers</td>
<td>Event triggers</td>
</tr>
</tbody>
</table>

### Levels

- **quiet**: Nothing
- **fatal**: Only fatal errors
- **error**: Errors and fatal errors
- **info**: Errors and general messages
- **verbose**: Errors and verbose messages
- **debug**: Errors, verbose and debug messages
- **debug2**: Errors, verbose and more debug messages
- **debug3**: Errors, verbose and even more debug messages
- **debug4**: Errors, verbose and even more debug messages
- **debug5**: Errors, verbose and even more debug messages

Don’t be above error

Don’t be below verbose if not troubleshooting

Partial list.
See manpage of `slurm.conf` for complete listing

Copyright 2019 SchedMD
www.schedmd.com
Logs - How to increase/decrease logs

- Changing Logging
  - All daemons and commands
    - Command Line: -vvv
  - All daemons
    - Edit .conf + reconfigure
  - scontrol setdebug LEVEL
  - scontrol schedloglevel LEVEL
  - scontrol setdebugflags [+|-]FLAG

**Use scontrol**

Increase logs, test and reduce logs.

Increase logging verbosity
> scontrol setdebug debug2
> scontrol setdebugflags +backfill
Wait for event...
Restore original logging levels
> scontrol setdebug info
> scontrol setdebugflags -backfill
Don't restart, reconfigure or SIGHUP slurmctld, slurmdbd or slurmd to drop and reopen logfiles.

Use SIGUSR2 for logrotate!
How to be sure that the ball is in Slurm’s court?

Slurm relies in a base setup. Although it works on degraded setups, a healthy setup also helps troubleshooting.
How to be sure that the ball is in Slurm’s court?

- Network issues
  - DNS
  - Firewalls
  - NTP (munge)

- Filesystem
  - StateSaveLocation
  - Unkillable

RPC plays a key role
How to be sure that the ball is in Slurm’s court?

- RAM usage on DB
  - Huge queries
  - Constant queries
  - Caching is not leaking

sdiag & stats to detect DoS!

MaxQueryTimeRange Archive & Purge
How to be sure that the ball is in Slurm’s court?

- **Reliable constraints:** cgroups
  - ConstrainDevices
  - ConstrainCores
  - ConstrainRAMSpace
  - ConstrainSwapSpace
  - ConstrainKmemSpace

- **JobAcctGatherParams/OverMemoryKill** is limited to JobAcctGatherFreq
- Demonized process in jobs escape from linuxproc
- **CUDA_VISIBLE_DEVICES** is NOT a constraint
- Avoid multiple cgroups controllers in the same mount
How to be sure that the ball is in Slurm’s court?

- Submit and Custom Plugins

  No Comment

  Have you tried cli_filter?
Why won’t my job run?
Why won’t my job run?

- To understand this let’s look at what happens in:
  - Scheduling / Backfill
Scheduling & Backfill

- This is not an all inclusive training on scheduling & backfill rather an introduction to help you, as the admin, troubleshoot issues.
Main scheduler (Backfill - Where it fits in)

- Slurm is designed to perform a quick and simple scheduling attempt at frequent intervals
  - At each job submission
  - At job completion on each of it's allocated nodes
  - At configuration changes
- Slower and more comprehensive scheduling attempts performed less frequently
Backfill

● The Backfill scheduler will start lower priority jobs if by doing so does not delay the expected start Time of any higher priority job (configurable exceptions)

● Setting accurate and reasonable run times is required for backfill to start lower priority jobs.
Backfill

- Making sure you have correctly tuned your SchedulerParameters is key when trying to push your sites workflow through optimally.
- Parameters such as:
Backfill

- Making sure you have correctly tuned your SchedulerParameters is key when trying to push your sites workflow through optimally.

- Parameters such as:
  - **bf_window**
    - The number of minutes into the future to look when considering jobs to schedule.
    - A value at least as long as the highest allowed time limit is generally advisable to prevent job starvation.
Backfill

- Making sure you have correctly tuned your SchedulerParameters is key when trying to push your sites workflow through optimally.
- Parameters such as:
  - bf_window
  - bf_continue
    - Setting this option will cause the backfill scheduler to continue processing pending jobs from its original job list after releasing locks even if job or node state changes.
Backfill

- Making sure you have correctly tuned your SchedulerParameters is key when trying to push your sites workflow through optimally.
- Parameters such as:
  - bf_window
  - bf_continue
  - bf_resolution
    - The number of seconds in the resolution of data maintained about when jobs begin and end.
Backfill

● Making sure you have correctly tuned your SchedulerParameters is key when trying to push your sites workflow through optimally.

● Parameters such as:
  ○ bf_window
  ○ bf_continue
  ○ bf_resolution

● And many other tunable parameters based on your needs.
  ○ See the slurm.conf documentation
Ensuring that your scheduler is configured correctly can fix or improve job throughput and help solve job starvation issues.
Why won’t my job run? - continued
Why won’t my job run? - continued

- There are enough resources available.
Why won’t my job run? - continued

- There are enough resources available.
- The cluster looks empty.
Why won’t my job run? - continued

- There are enough resources available.
- The cluster looks empty.
- Do I have the right permissions to run?
Why won’t my job run? - continued

- One common question we see is regarding a job that a user believes should run on a node that is already partially allocated.
Why won’t my job run? - continued

Example: CPU with 4 cores

gres.conf:

Name=gpu Type=p100 File=/dev/nvidia0 COREs=0,1
Name=gpu Type=gpu File=/dev/nvidia1 COREs=2,3

Job 123 is on cores 0,1
Job 124 needs the p100
As soon as some reason is found why a job cannot be started, that is recorded in the job's “reason” field and the scheduler moves on to the next job. Some common reasons why jobs are pending:

<table>
<thead>
<tr>
<th>Reason</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Resources being reserved for higher priority job</td>
</tr>
<tr>
<td>Resources</td>
<td>Required resources are in use</td>
</tr>
<tr>
<td>Dependency</td>
<td>Job dependencies not yet satisfied</td>
</tr>
<tr>
<td>Reservation</td>
<td>Waiting for advanced reservation</td>
</tr>
<tr>
<td>AssociationJobLimit</td>
<td>User or account job limit reached</td>
</tr>
<tr>
<td>AssociationResourceLimit</td>
<td>User or account resource limit reached</td>
</tr>
<tr>
<td>AssociationTimeLimit</td>
<td>User or account time limit reached</td>
</tr>
<tr>
<td>QOSJobLimit</td>
<td>Quality Of Service (QOS) job limit reached</td>
</tr>
<tr>
<td>QOSResourceLimit</td>
<td>Quality Of Service (QOS) resource limit reached</td>
</tr>
<tr>
<td>QOSTimeLimit</td>
<td>Quality Of Service (QOS) time limit reached</td>
</tr>
</tbody>
</table>
$ scontrol show job 21543
JobId=21543 JobName=wrap
UserId=agil(1000) GroupId=agil(1000) MCS_label=N/A
Priority=251879 Nice=0 Account=development QOS=normal
JobState=PENDING Reason=Dependency
Requeue=1 Restarts=0 BatchFlag=1 Reboot=0
Runtime=0:00:00 TimeLimit=UNLIMITED
SubmitTime=2019-09-02T18:08:37 EligibleTime=Unknown
AccrueTime=2019-09-02T18:08:37
StartTime=Unknown EndTime=Unknown Deadline=N/A
SuspendTime=None SecsPreSuspend=0 LastSchedEval=2019-09-02T18:08:37
Partition=batch AllocNode:Sid=agil-work_station:22616
ReqNodeList=(null) ExcNodeList=(null)
NodeList=(null) NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1
ReqB:S:C:T=0:0:*:* TRES=cpu=1,mem=512M,node=1,billing=1
Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=* MinCPUsNode=1 MinMemoryCPU=512M MinTmpDiskNode=0
Features=(null) DelayBoot=00:00:00 OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
Command=(null) WorkDir=/home/agil/workspace/slurm/bugs/7468
Stderr=/home/agil/workspace/slurm/bugs/7468/slurm-21543.out
Stdin=/dev/null StdOut=/home/agil/workspace/slurm/bugs/7468/slurm-21543.out
Power=

$ scontrol show job 26315
JobId=26315 JobName=wrap
UserId=jason(1000) GroupId=jason(1000) MCS_label=bio
Priority=25416 Nice=0 Account=bio QOS=normal
JobState=PENDING Reason=AssocMaxJobsLimit
Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
Runtime=0:00:00 TimeLimit=00:02:00
AccrueTime=2019-09-12T13:50:39
StartTime=Unknown EndTime=Unknown Deadline=N/A
SuspendTime=None SecsPreSuspend=0 LastSchedEval=2019-09-12T13:50:40
Partition=main AllocNode:Sid=nh-grey:6638
ReqNodeList=(null) ExcNodeList=(null)
NodeList=(null) NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1
ReqB:S:C:T=0:0:*:* TRES=cpu=1,node=1,billing=1
Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=* MinCPUsNode=1 MinMemoryCPU=512M MinTmpDiskNode=0
Features=(null) DelayBoot=00:00:00 OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
Command=(null) WorkDir=/home/jason/slurm/master
Stderr=/home/jason/slurm/master/slurm-26315.out
Stdin=/dev/null StdOut=/home/jason/slurm/master/slurm-26315.out
Power=
Why won’t my job run? - continued

- We continue to improve logging (for example)
  - Logs for BadConstraints have been improved see the following.
    - [https://github.com/SchedMD/slurm/commit/20c2b6151d6d10fee7e012555d30f9fd529b7dc3](https://github.com/SchedMD/slurm/commit/20c2b6151d6d10fee7e012555d30f9fd529b7dc3)
Bugzilla - Consider the following

- Security
- Proper data / attachments
- Ticket visibility
Bugzilla is by default: open to the internet
Make sure to avoid uploading configs that contain passwords such as slurmd.db.conf (scrub the password).
Attachments and comments can be marked as private by SchedMD
Problem Reporting - Proper data / attachments

- Report using bugzilla
  - https://bugs.schedmd.com/
- Identify Slurm version
- Provide configuration files
  - Essential to identify the plugins used
- Provide logs demonstrating the problems
- Make sure you compress large files

Practically perfect in every way!

Copyright 2019 SchedMD
www.schedmd.com
Problem Reporting - Ticket Visibility

- When needed SchedMD can limit the visibility of comments, attachments and the entire bug if needed.
References

- https://slurm.schedmd.com/troubleshoot.html
- https://slurm.schedmd.com/faq.html
- https://slurm.schedmd.com/quickstart_admin.html (#upgrade)

- https://slurm.schedmd.com/SLUG18/field_notes2.pdf
- https://slurm.schedmd.com/SLUG17/FieldNotes.pdf

- https://bugs.schedmd.com
- http://groups.google.com/group/slurm-users