



All Things TRES (Trackable RESources) SC15

Brian Christiansen
brian@schedmd.com

Copyright 2015 SchedMD LLC
<http://www.schedmd.com>

Overview

- Need
- Setup
- Transition
- Reporting
- Fairshare
- Priority

Need



- Limits on more resources other than CPU/Memory/Nodes
 - GRES, Licenses, etc
- Method for accounting what resources were really used
 - Other than just cpu
- Easier way to add more limits (for developers)
 - No database alteration needed for future TRES

Setup

- All TRES are global and are defined in the `slurm.conf`
 - Available to all clusters
- AccountingStorageTRES
 - Used to define which TRES are to be tracked on the system. By default CPU, Energy, Memory, and Node are tracked. This will be the case whether specified or not.
 - `AccountingStorageTRES=<TRES Type>[/<name>[:<subname>]]`
 - Example
 - `AccountingStorageTRES=gres/gpu:tesla,license/iop1,bb/cray`

Transition

- sacctmgr
 - [Grp|Max] [cpu|mem|node]* limits now [Grp|Max]TRES*
 - GrpTRES=cpu=500,mem=10000,nodes=100
 - (GrpCpus=500 GrpMem=10000 GrpNodes=100)
 - Old definitions still work, for legacy scripts
 - -1 still how to remove limits GrpTRES=cpu=-1,mem=-1,nodes=1000

Transition

- sacctmgr
 - New/Extended Association|QOS options (all work for any TRES)
 - GrpTRES
 - GrpTRESMins
 - GrpTRESRunMins
 - MaxTRESPerJob
 - MaxTRESPerNode
 - MaxTRESPerUser*
 - MaxTRESMinsPerJob
 - MinTRESPerJob*

*only applicable to QOS

Transition

- `scontrol/squeue/sacct`
 - Can display TRES as well
 - ex. `squeue -O jobid,name,tres`
 - When a limit is reached the reason field in a job has a unique reason for each TRES type/limit combo
 - `QOSGrpCpuLimit`
 - `QOSGrpMemLimit`
 - `AssocGrpCpuLimit`
 - `AssocGrpMemLimit`
 - etc.

Reporting

- sreport
 - Previously would only report on CPU utilization
 - Now can report on **any** TRES (except Node)

Reporting

- Need more memory? Or less cpus?

```
$ sreport -tminper cluster utilization --tres="cpu,mem" start=2015-09-01T00:00:00
```

```
-----  
Cluster Utilization 2015-09-01T00:00:00 - 2015-09-01T23:59:59
```

```
Use reported in TRES Minutes/Percentage of Total
```

```
-----  
Cluster      TRES Name      Allocated      Down      PLND Down      Reserved      Idle      Reported  
-----  
  compy      cpu      253440(20.00%)      0(0.00%)      0(0.00%)      0(0.00%)      1013760(80.00%)      1267200(100.00%)  
  compy      mem      4582306080(90.00%)      0(0.00%)      0(0.00%)      509145120(10.00%)      0(0.00%)      5091451200(100.00%)  
-----
```

Reporting

- GPUs being used?

```
$ sreport -tminper cluster utilization --tres="cpu,gres/gpu" start=2015-09-02T00:00:00
```

```
-----  
Cluster Utilization 2015-09-02T00:00:00 - 2015-09-2T23:59:59
```

```
Use reported in TRES Minutes/Percentage of Total
```

```
-----  
Cluster      TRES Name      Allocated      Down      PLND Down      Reserved      Idle      Reported  
-----  
compy        cpu            1140480(90.00%)  0(0.00%)  0(0.00%)      126720(10.00%)  0(0.00%)  1267200(100.00%)  
compy        gres/gpu       63360(20.00%)   0(0.00%)  0(0.00%)      0(0.00%)      253440(80.00%)  316800(100.00%)  
-----
```

Reporting

- Which GPUs are being used most?

```
$ sreport -tminper cluster utilization --tres="gres/gpu:k40,gres/gpu:k80" start=2015-09-02T00:00:00
```

```
-----  
Cluster Utilization 2015-09-02T00:00:00 - 2015-09-2T23:59:59
```

```
Use reported in TRES Minutes/Percentage of Total
```

```
-----  
Cluster      TRES Name      Allocated      Down      PLND Down      Reserved      Idle      Reported  
-----  
  compy    gres/gpu:k40    63360(20.00%)  0(0.00%)  0(0.00%)      0(0.00%)    253440(80.00%)  316800(100.00%)  
  compy    gres/gpu:k80    190080(60.00%)  0(0.00%)  0(0.00%)      0(0.00%)    126720(40.00%)  316800(100.00%)  
-----
```

Fairshare

- Previously only total cpus was accounted for in fairshare
- If a job used 1 CPU and all the memory on the machine the job was only charged for 1 CPU when it really used the whole node
- Now, any TRES can be accounted for in fairshare
 - TRESBillingWeights

Fairshare

- TresBillingWeights configured per partition
- Billing weights are specified as a comma-separated list of <TRES Type>=<TRES Billing Weight> pairs
- TRESBillingWeights=CPU=1.0,Mem=0.25G,GRES/gpu=2.0
- Two methods of calculating billable TRES
 - MAX_TRES
 - SUM of TRES

Fairshare

- SUM of TRES
 - Default
 - $\text{SUM}(\langle \text{TRES} \rangle * \langle \text{TRES Weight} \rangle, \dots)$
 - Good if you want to account for what you are using
- MAX_TRES
 - $\text{PriorityFlags} = \text{MAX_TRES}$
 - $\text{MAX}(\text{Node TRES}) + \text{SUM}(\text{Global TRES})$
 - Good if you want to account for any one resource that is blocking other jobs from running on a node

Fairshare

- TRESBillingWeights=CPU=1.0,Mem=0.25G
- 16CPU, 64GB nodes

SUM of TRES:

	CPU	Mem	
Job1:	$(1 * 1.0)$	$(60 * 0.25)$	$= (1 + 15) = 16$
Job2:	$(16 * 1.0)$	$(1 * 0.25)$	$= (16 + 0.25) = 16.25$
Job3:	$(16 * 1.0)$	$(60 * 0.25)$	$= (16 + 15) = 31$

MAX_TRES:

	CPU	Mem	
Job1:	$\text{MAX}((1 * 1.0), (60 * 0.25))$		$= 15$
Job2:	$\text{MAX}((16 * 1.0), (1 * 0.25))$		$= 16$
Job3:	$\text{MAX}((16 * 1.0), (64 * 0.25))$		$= 16$

Priority

- PriorityWeightTRES
 - List of TRES Types and weights
 - PriorityWeightTRES=CPU=1000,Mem=2000,GRES/gpu=3000
- Control how much a TRES contributes to the job's priority
- Node TRES (i.e. CPU, Mem, GRES, Node) are normalized against total TRES configured in a partition
- Global TRES (i.e. license, bb) are normalized against the global amount in the system

Priority - Example

- AccountingStorageTRES=cpu,mem,gres/gpu
- PriorityWeightTRES=cpu=1000,gres/gpu=3000
- If a partition has 80 cpus and a job uses 8, then the priority factor is .1 (or 10%)

```
$ sprio
```

JOBID	PRIORITY	AGE	FAIRSHARE	TRES
3	625	0	500	cpu=125
5	600	0	500	cpu=100
6	812	0	500	cpu=12,gres/gpu=300

Conclusion



- More possibilities for limits
- Ability to report on different resource utilization
- Customizable fairshare prioritization
- Finer grained priority calculations

Questions?

