Profile with HDF5

Slurm 2013 User Group

Danny Auble, SchedMD
Yiannis Georgiou, Bull
Rod Schultz, Bull

What is Profiling With HDF5?

Detailed collection of performance data of a parallel job

- More detail than can reasonably be stored in an accounting database
- Data from all tasks on all nodes consolidated in one (HDF5) dataset
- Controls to limit data collection to only a few jobs to minimize overhead on the entire system

Why Profile?

- Profiling has traditionally been used to improve an applications use of resources, particularly CPUs
- There is an increasing need to improve the scheduling and placement of an application on the resources of the supercomputer
- It is now important to schedule applications to efficiently use energy and air conditioning
- It is also important to allocate resources that are physically close together to minimize network latency for both message passing and use of parallel file systems

How to Profile in Slurm

There’s a ***plugin*** for that!

How to Profile in Slurm …

Actually several plugins

- A ***Framework*** plugin that provides an API for writing to an HDF5 file and other supporting infrastructure
- ***Data Gathering*** plugins that interact with other parts of Slurm, the operating system, or drivers for hardware and software sensors

AcctGatherProfile Framework Plugin

- The **AcctGatherProfileType/HDF5** plugin allows Slurm to coordinate collecting data on jobs it runs on a cluster
- The data comes from data gathering plugins that periodically sample various performance data either collected by Slurm, the operating system, or component software
- These plugins call the framework to **add a sample**
- The plugin will record the data from each source as a **Time Series** and accumulate totals for each statistic for the job.

AcctGatherProfile Framework Plugin …

- The plugin provides an API to store a periodic sample
- The API defines **known** data structure types that callers fill with data. The plugin has a library that reads and writes those types in the HDF5 file and perform common generic operations
  - type implementation is similar to plugin implementation
- i.e. the Plugin does gather the data itself but expects other plugins to pass a known data structure through the API

API Example

An Example of a **known** data type

(defined in the ***framework*** plugin -- acct_gather_profile/hdf5/hdf5_api.h)

```
typedef struct acct_energy_data {
   time_t   time;
   uint64_t power;
   uint64_t cpu_freq;
} acct_energy_data_t;
```

Primary Call to Profile API

(Called from a ***data collection*** plugin -- acct_gather_energy plugin)

```
acct_energy_data_t ener;
ener.cpu_freq = 1;
ener.time = time(NULL);
ener.power = local_energy->current_watts;
acct_gather_profile_g_add_sample_data(ACCT_GATHER_PROFILE_ENERGY, &ener);
```

You can't arbitrarily add any type of data

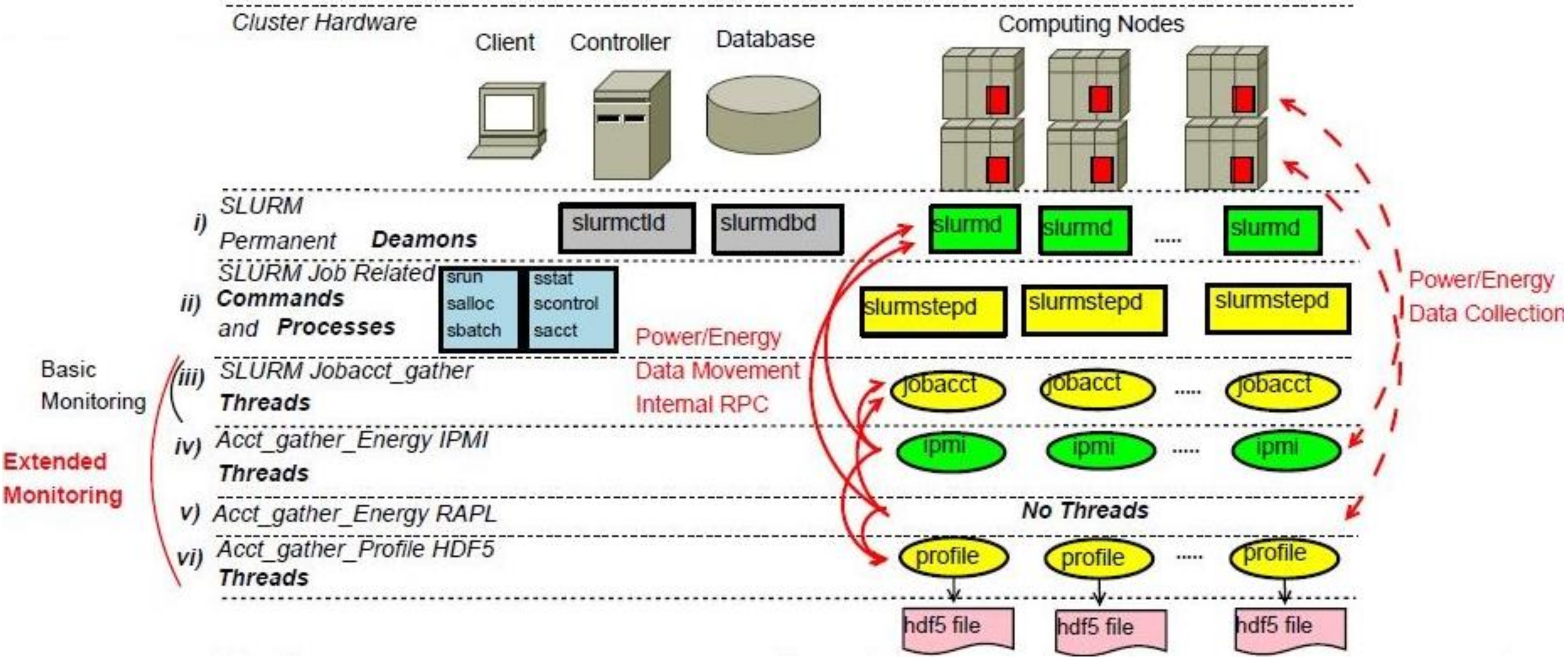
- A structure must be defined in the plugin api
- Functions to transform the structure into HDF5 objects must be implemented

AcctGather Data Gather Plugins

- These plugins gather data and call the Framework plugin API

Type	Implementation	Data
AcctGatherEnergy	ipmi	Energy
AcctGatherEnergy	rapl	Energy
AcctGatherFilesystem	lustre	Parallel Filesystem
AcctGatherInfiniband	ofed	Network
JobAcctGather	linux	cpu, memory, disk

Profile Plugin Architecture



After the job terminates, sh5util is used to merge the node step HDF5 files into a job HDF5 file

Data Flow

- While a job executes, the data collection plugins are periodically called on each node (by Slurmstepd)
- They in turn call the framework plugin to add_sample
- The Profile plugin stores the data in a ***node-step*** HDF5 file on a shared file system
- When the job ends, sh5util merges all the node-step files into one ***job*** HDF5 file (This isn't automatic but is often done as an additional sbatch in an sbatch script)
- sh5util can also extract subsets of data as a text file to be imported into other analysis tools such as spreadsheets

HDF5

The HDF Group

www.hdfgroup.org

What is HDF5?

- A system widely used in HPC supporting structured data.
- **Has a versatile data model** that can represent very complex data objects and a wide variety of metadata
- **Has a completely portable file format** with no limit on the number or size of data objects stored
- **Has an open source software library** that runs on a wide range of computational platforms, from cell phones to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces
- **A rich set of integrated performance features** that allow access time and storage space optimizations
- **Tools and applications** for managing, manipulating, viewing, and analyzing the data in the collection

HDF5 File Structure

- The internal structure of a HDF5 file resembles a file system with **groups** being similar to *directories* and **data sets** being similar to *files*
- A data set is a multi-dimensional array of elements with supporting metadata
- **Attributes** can be attached to groups to store application defined properties

Profiler Use of HDF5 Structure

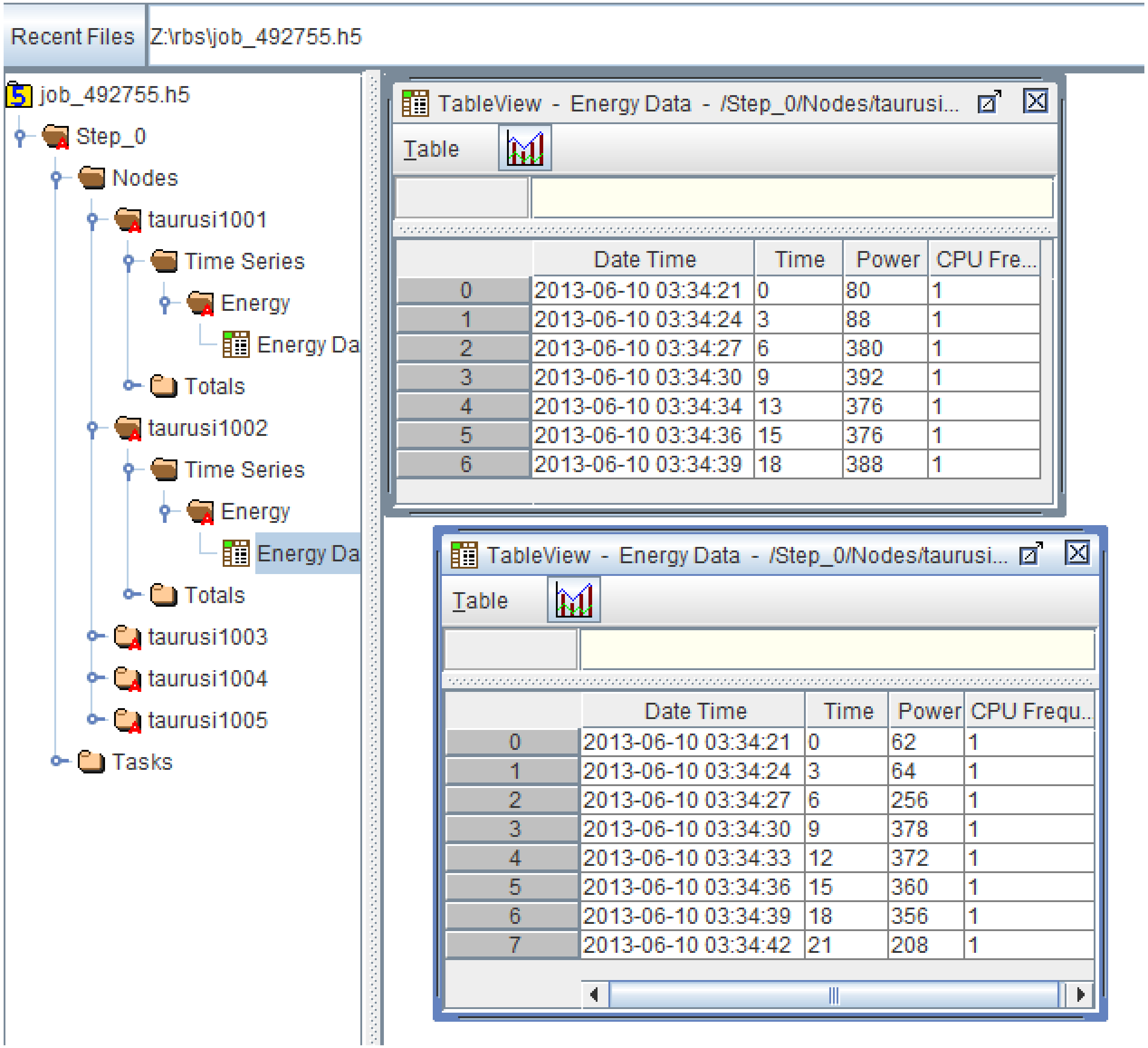
- In the **job** file, there will be a *group* for each **step** of the job
- Within each step, there will be a *group* for **Nodes**, and a *group* for **Tasks**
  - The **Nodes** group will have a *group* for each **node** in the step allocation
    - For each node group, there is a *group* for **Time Series** and another for **Totals**
      - The *Time Series* group contains a *group/dataset* containing the time series for each data type collected
      - The *Totals* group contains a corresponding *group/dataset* that has the Minimum, Average, Maximum, and Sum Total for each item in the time series
  - The **Tasks** group will only contain a *group* for each task. It primarily contains an attribute stating the node on which the task was executed. This set of groups is essentially a cross reference table.

HDFView

- **HDFView** is a visual tool for browsing and editing HDF4 and HDF5 files. Using HDFView, you can view a file hierarchy in a tree structure.

- http://www.hdfgroup.org/hdf-java-html/hdfview/

HDFView example



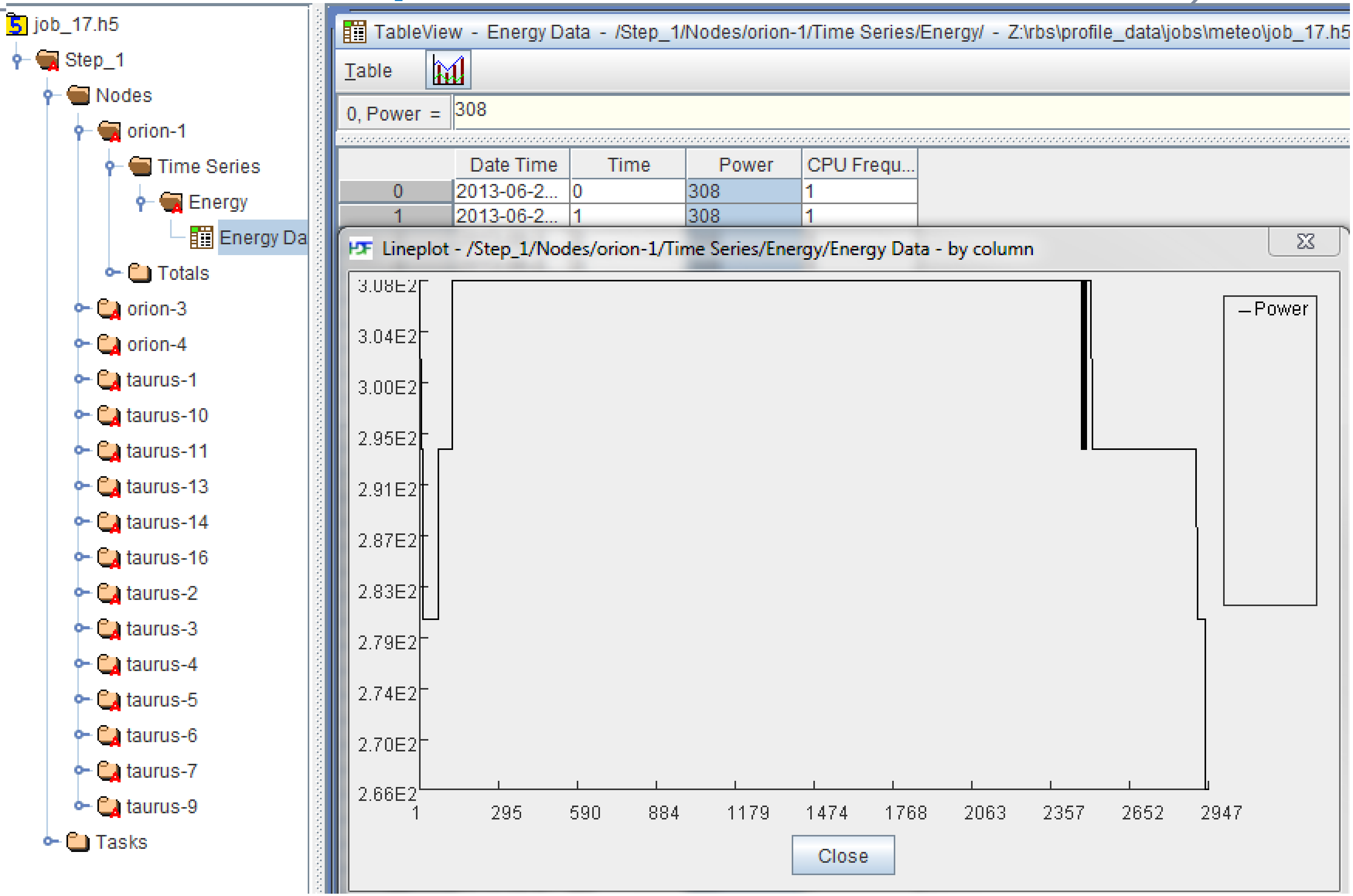
TableView - Energy Data - /Step_0/Nodes/taurusi...

	Date Time	Time	Power	CPU Fre...
0	2013-06-10 03:34:21	0	80	1
1	2013-06-10 03:34:24	3	88	1
2	2013-06-10 03:34:27	6	380	1
3	2013-06-10 03:34:30	9	392	1
4	2013-06-10 03:34:34	13	376	1
5	2013-06-10 03:34:36	15	376	1
6	2013-06-10 03:34:39	18	388	1

TableView - Energy Data - /Step_0/Nodes/taurusi...

	Date Time	Time	Power	CPU Frequ...
0	2013-06-10 03:34:21	0	62	1
1	2013-06-10 03:34:24	3	64	1
2	2013-06-10 03:34:27	6	256	1
3	2013-06-10 03:34:30	9	378	1
4	2013-06-10 03:34:33	12	372	1
5	2013-06-10 03:34:36	15	360	1
6	2013-06-10 03:34:39	18	356	1
7	2013-06-10 03:34:42	21	208	1

HDFView Graph



TableView - Energy Data - /Step_1/Nodes/orion-1/Time Series/Energy/ - Z:\rbs\profile_data\jobs\meteo\job_17.h5

0, Power = 308

	Date Time	Time	Power	CPU Frequ...
0	2013-06-2...	0	308	1
1	2013-06-2...	1	308	1

Lineplot - /Step_1/Nodes/orion-1/Time Series/Energy/Energy Data - by column

Profiling Jobs

Profiling Jobs

- ***Data Collection***

Data collection happens while a step is running. Various plugins periodically sample counters or sensors and call the framework to add the sampled data into a HDF5 file for the step. The plugins run inside the slurmd daemon so there is one step file on each node. (As noted before, these files are on a shared file system.)

The **--profile** option on *salloc* | *sbatch* | *srun* controls whether data is collected and what type of data is collected. The following options are currently available.

All	All data types are collected.
None	No data types are collected. This is the default.
Energy	Energy data is collected.
Lustre	Lustre I/O data is collected.
Network	Network (InfiniBand) data is collected.
Task	Task (I/O, Memory, ...) data is collected.

The **--acctg-freq** option may be used to override the JobAcctGatherFrequency parameter in slurm.conf. Different sample rates can be specified for each data type

```
e.g.      srun -N2 -n3 --profile=task,energy --acctg-freq=energy=3,task=60 myjob
```

Profiling Jobs ...

- **Data Consolidation**

  The node-step files are merged into one HDF5 file for the job using the **sh5util** program. They are then deleted.

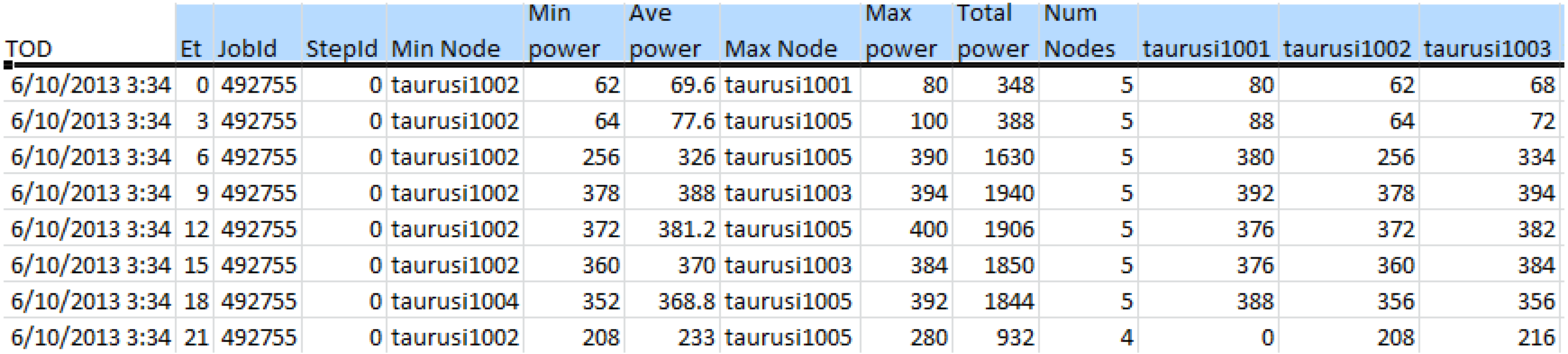
  e.g. `sbatch -n1 -d$Slurm_JOB_ID --wrap="sh5util -j $Slurm_JOB_ID"`

- **Data Extraction**

  The **sh5util** program can also extract all samples for a specific data item from a time series and write a comma separated value (csv) file for importation into other analysis tools such as spreadsheets.

  e.g. `sh5util -j 42 –item-extract --series=Energy --data=power`

csv Output in Spreadsheet



					Min	Ave		Max	Total	Num			
TOD	Et	JobId	StepId	Min Node	power	power	Max Node	power	power	Nodes	taurusi1001	taurusi1002	taurusi1003
6/10/2013 3:34	0	492755	0	taurusi1002	62	69.6	taurusi1001	80	348	5	80	62	68
6/10/2013 3:34	3	492755	0	taurusi1002	64	77.6	taurusi1005	100	388	5	88	64	72
6/10/2013 3:34	6	492755	0	taurusi1002	256	326	taurusi1005	390	1630	5	380	256	334
6/10/2013 3:34	9	492755	0	taurusi1002	378	388	taurusi1003	394	1940	5	392	378	394
6/10/2013 3:34	12	492755	0	taurusi1002	372	381.2	taurusi1005	400	1906	5	376	372	382
6/10/2013 3:34	15	492755	0	taurusi1002	360	370	taurusi1003	384	1850	5	376	360	384
6/10/2013 3:34	18	492755	0	taurusi1004	352	368.8	taurusi1005	392	1844	5	388	356	356
6/10/2013 3:34	21	492755	0	taurusi1002	208	233	taurusi1005	280	932	4	0	208	216

Performance Impacts

Performance – File Size

- Two Files Types;
  - Node-step file, (one file for each node for each step)
  - Job File (merge of all node-step files)

- Node-step size primarily dependent on number of samples
  - Each HDF5 Group is 1500-2000 bytes
  - Each sample is an HDF5 Group plus 100-200 bytes of data.
  - Node-step of energy data, 3000 samples is ~5mb.
  - (estimate of 1 second sample for 1 day is ~150mb / node)
  - They have to be read over the management network, but are deleted after the merge.
  - At the maximum sample rate this generates 2kb/sec/node on the NFS server.

Performance – File Size …

- Job file size primarily dependent on number of nodes and number of samples.
  - Each node is a HDF5 Group + 2 Groups per series.
  - Collected data (50-100 bytes) * number of samples
  - Job on 16 nodes of energy data, 3000 samples is ~3mb.
  - Job on 260 nodes of energy data, 10 samples is ~4.5mb
  - Trivial job is ~20kb

Performance – File Size …

More examples (Task Data)

Samples	Nodes	Tsk/Nd	Tasks	NodeStep Size(kb)	Job Size(kb)
101	1	1	1	217	26
101	2	1	2	217	45
101	4	1	4	217	83
101	8	1	8	217	160
101	16	1	16	217	315
101	32	1	32	217	624
51	1	1	1	114	26
51	2	1	2	114	45
51	4	1	4	114	64
51	8	1	8	114	123
51	1	2	2	220	33
51	2	2	4	220	58
51	2	4	8	435	104
51	4	2	8	220	110

Performance – Compute nodes (CPU)

- Primary factor is sample rate
- Lustre, Network and Task data are fast as all data is either in Slurm data structures or in the /proc directory. Sample rates of every 60 seconds is adequate for most needs.
- Energy has more overhead as another thread is used and sample rates are typically every few seconds.
- While sampling task data every second for 24 hours, there was almost no increase is Slurmstepd cpu time.

Performance – Compute nodes (Memory)

- There is an increase in RSS for Slurmstepd, but it levels off
- RSS seems to increase until it hits around 30mb

Performance – Elapsed Time

- Application effect is CPU and IO as above
- Merge of node-step files into job file
  - One task on one node of the allocation is typically used when the job ends to do the merge.
  - The merge is a serial process, one node-step file at a time is read and added to the job file.
  - Is affected by size of node-step files, and speed of shared file system.
  - Merge of a job that collected energy data 1/sec for 10 minutes on 270 nodes took 3 minutes. (1.5 nodes/sec)

Administration

Administration

Shared File System

The HDF5 Profile Plugin requires a common shared file system on all the compute nodes. While a job is running, the plugin writes a file into this file system for each step of the job on each node. When the job ends, the merge process is launched and the node-step files are combined into one HDF5 file for the job.

- The root of the directory structure is declared in the **ProfileHDF5Dir** option in the **acct_gather.conf** file. The directory will be created by Slurm if it doesn't exist.
- Each user that creates a profile will have a subdirectory to the profile directory that has read/write permission only for the user.

Administration ...

- Configuration parameters

  The profile plugin is enabled in the **slurm.conf** file, but is internally configured in the **acct_gather.conf** file.

slurm.conf parameters

- **AcctGatherProfileType**=acct_gather_profile/hdf5 enables the HDF5 Profile Plugin
- **JobAcctGatherFrequency**={energy=freq {,lustre=freq {,network=freq , {task=freq}}}} sets default sample frequencies for data types.
- One or more of the following plugins must also be configured.
  - AcctGatherEnergyType=acct_gather_energy/ipmi
  - AcctGatherEnergyType=acct_gather_energy/rapl
  - AcctGatherFilesystemType=acct_gather_filesystem/lustre
  - AcctGatherInfinibandType=acct_gather_infiniband/ofed
  - JobAcctGatherType=job_acct_gather/linux

Administration ...

- Configuration parameters

acct_gather.conf parameters

There are parameters directly used by the HDF5 Profile Plugin.

ProfileHDF5Dir=path is the path to the shared folder into which the acct_gather_profile plugin will write detailed data as an HDF5 file. The directory is assumed to be on a file system shared by all compute nodes. This is a required parameter.

ProfileHDF5CollectDefault=opt{,opt{,opt}} is the default –profile=< value> for data types collected for each job submission. It is a comma separated list of data streams. Use this option with caution. A node-step file will be created for every node of every step for every job. They will not automatically be merged into job files. (Even job files for small jobs would fill the file system.) This option is intended for test environments where you might want to profile a series of jobs but do not want to have to add the --profile option to the launch scripts.

Other acct_gather plugins may have their own parameters. See Slurm documentation for details.

Sample conf files

slurm.conf

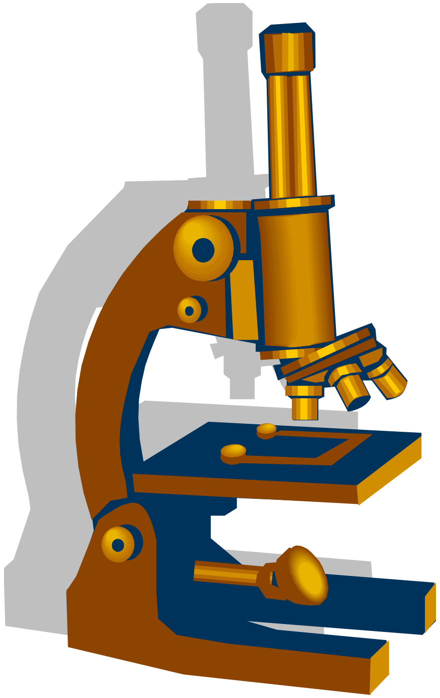
```
DebugFlags=Profile
AcctGatherProfileType=acct_gather_profile/hdf5

JobAcctGatherType=jobacct_gather/linux
JobAcctGatherFrequency=energy=5,lustre=60,network=60,task=60
AcctGatherEnergyType=acct_gather_energy/ipmi
AcctGatherFilesystemType=acct_gather_filesystem/lustre
AcctGatherInfinibandType=acct_gather_infiniband/ofed
```

acct_gather.conf

```
# Parameters for AcctGatherEnergy/ipmi plugin
EnergyIPMIFrequency=10
EnergyIPMICalcAdjustment=yes
#
# Parameters for AcctGatherProfileType/hdf5 plugin
ProfileHDF5Dir=/app/Slurm/profile_data
# Parameters for AcctGatherInfiniband/ofed plugin
InfinibandOFEDFrequency=4
InfinibandOFEDPort=1
```

Appendices



Energy Data

- AcctGatherEnergyType=acct_gather_energy/ipmi is required in slurm.conf to collect energy data.
- JobAcctGatherFrequeny=Energy=<freq> should be set in either slurm.conf or via –acctg-freq command line option.

The IPMI energy plugin also needs the EnergyIPMIFrequency value set in the acct_gather.conf file. This sets the rate at which the plugin samples the external sensors. This value should be the same as the energy=sec in either JobAcctGatherFrequency or --acctg-freq.

Note that the IPMI and profile sampling is not synchronous. The profile sample simply takes the last available IPMI sample value. If the profile energy sample is more frequent than the IPMI sample rate, the IPMI value will be repeated. If the profile energy sample is greater than the IPMI rate, IPMI values will be lost.

Also note that smallest effective IPMI (EnergyIPMIFrequency) sample rate for 2013 era Intel processors is 3 seconds.

Note that Energy data is collected for the entire node so it is only meaningful for exclusive allocations.

- Each data sample in the Energy Time Series contains the following data items.

Date Time	Time of day at which the data sample was taken. This can be used to correlate activity with other sources such as logs.
Time	Elapsed time since the beginning of the step.
Power	Power consumption during the interval.
CPU Frequency	CPU Frequency at time of sample in kilohertz.

Lustre Data

- AcctGatherFilesystemType=acct_gather_filesystem/lustre is required in Slurm.conf to collect lustre data.
- JobAcctGatherFrequeny=Lustre=<freq> should be set in either Slurm.conf or via –acctg-freq command line option.

- Each data sample in the Lustre Time Series contains the following data items.

Date Time	Time of day at which the data sample was taken. This can be used to correlate activity with other sources such as logs.
Time	Elapsed time since the beginning of the step.
Reads	Number of read operations.
MegabytesRead	Number of megabytes read.
Writes	Number of write operations.
MegabytesWrite	Number of megabytes written.

Network (Infiniband) Data

- AcctGathertInfinibandType=acct_gather_infiniband/ofed is required in Slurm.conf to collect Network data.
- JobAcctGatherFrequeny=Network=<freq> should be set in either Slurm.conf or via –acctg-freq command line option.
- Each data sample in the Network Time Series contains the following data items.

Date Time Time of day at which the data sample was taken.
This can be used to correlate activity with other sources such as logs.

Time Elapsed time since the beginning of the step.

PacketsIn Number of packets coming in.

MegabytesIn Number of megabytes coming in through the interface.

PacketsOut Number of packets going out.

MegabytesOut Number of megabytes going out through the interface.

Task Data

- JobAcctGatherType=jobacct_gather/linux is required in Slurm.conf to collect task data
- JobAcctGatherFrequeny=Task=<freq> should be set in either Slurm.conf or via –acctg-freq command line option.

  The frequency should be set to at least 30 seconds for CPU utilization to be meaningful (since the resolution of cpu time in linux is 1 second)

- Each data sample in the Task Time Series contains the following data items.

Date Time Time of day at which the data sample was taken.
This can be used to correlate activity with other sources such as logs.
Time Elapsed time since the beginning of the step.
CPUFrequency CPU Frequency at time of sample.
CPUTime Seconds of CPU time used during the sample.
CPUUtilization CPU Utilization during the interval.
RSS Value of RSS at time of sample.
VMSize Value of VM Size at time of sample.
Pages Pages used in sample.
ReadMegabytes Number of megabytes read from local disk.
WriteMegabytes Number of megabytes written to local disk.

Questions

More info:
Man pages for slurm.conf, acct_gather.com, sh5util
http://slurm.schedmd.com/hdf5_profile_user_guide.html

Our email:
da@schedmd.com
Yiannis.Georgiou@exchange.bull.net
Rod.Schultz@bull.com (now)
Rod.Schultz@exchange.bull.net (coming soon)