Cray XC Power Monitoring and Management Tutorial

CUG-2015:

Steven J. Martin ..... (stevem@cray.com)

David Rush ..... (rushd@cray.com)

Matthew Kappel ..... (mkappel@cray.com)

Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

XC PM Tutorial: Outline

- Cray XC Power Monitoring and Control Overview
- Hardware power monitoring
- Plotting system data
- SQL and PMDB
- CAPMC
- Prototype Application
- RUR
- xtpmaction
- Backup

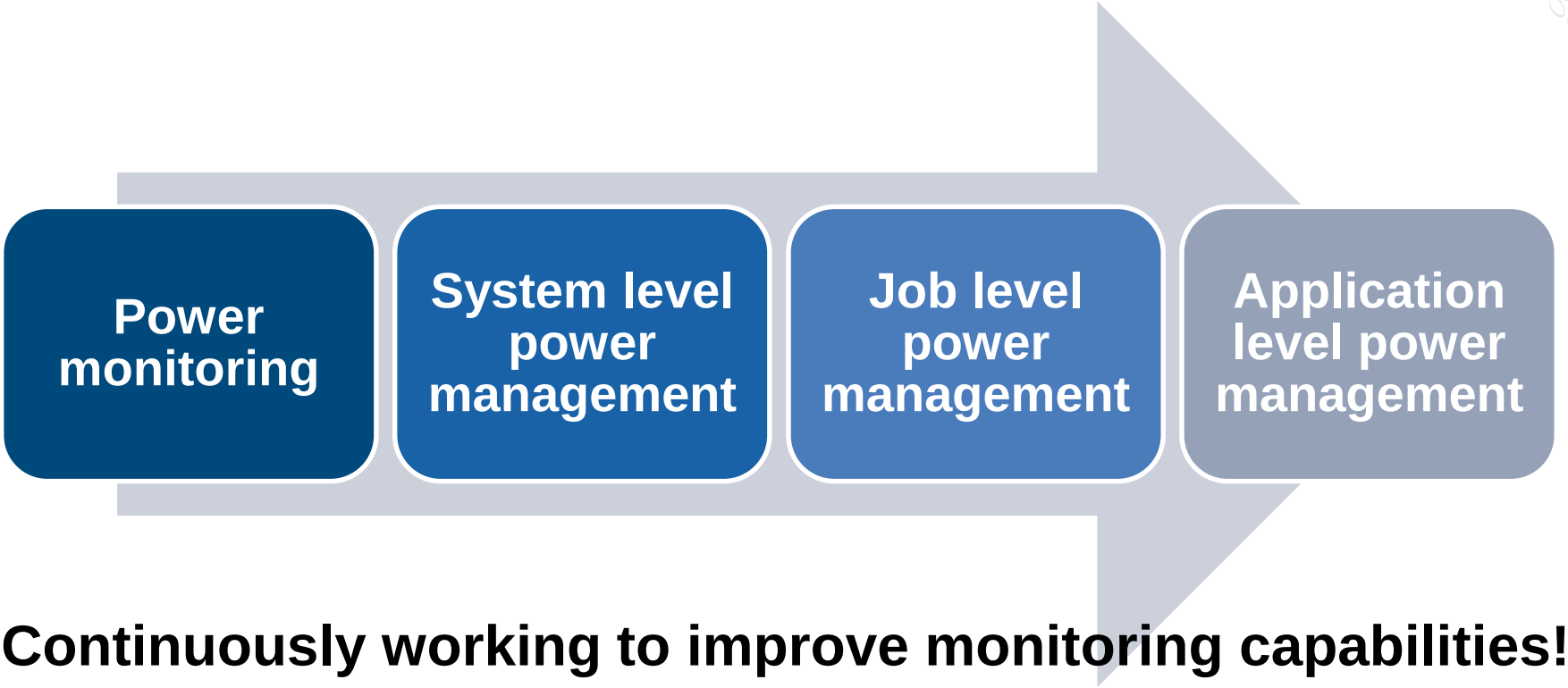
XC PM Overview

Power Management: Motivation & Philosophy

- **Motivation**
  - System procurements are increasingly constrained
    - Site power & cooling limitations
    - Cost of system power and cooling
  - Customer requirements
    - Power monitoring tools
    - Management of power consumption
    - Better performance per watt
  - Power limitations
    - 20 MW max power target for extreme-scale systems of the future

- **Philosophy**
  - Do not waste energy!
  - Measure power, so you know where it is going
  - Allow customers to affect greater power savings

Power Management: Progression



Continuously working to improve monitoring capabilities!

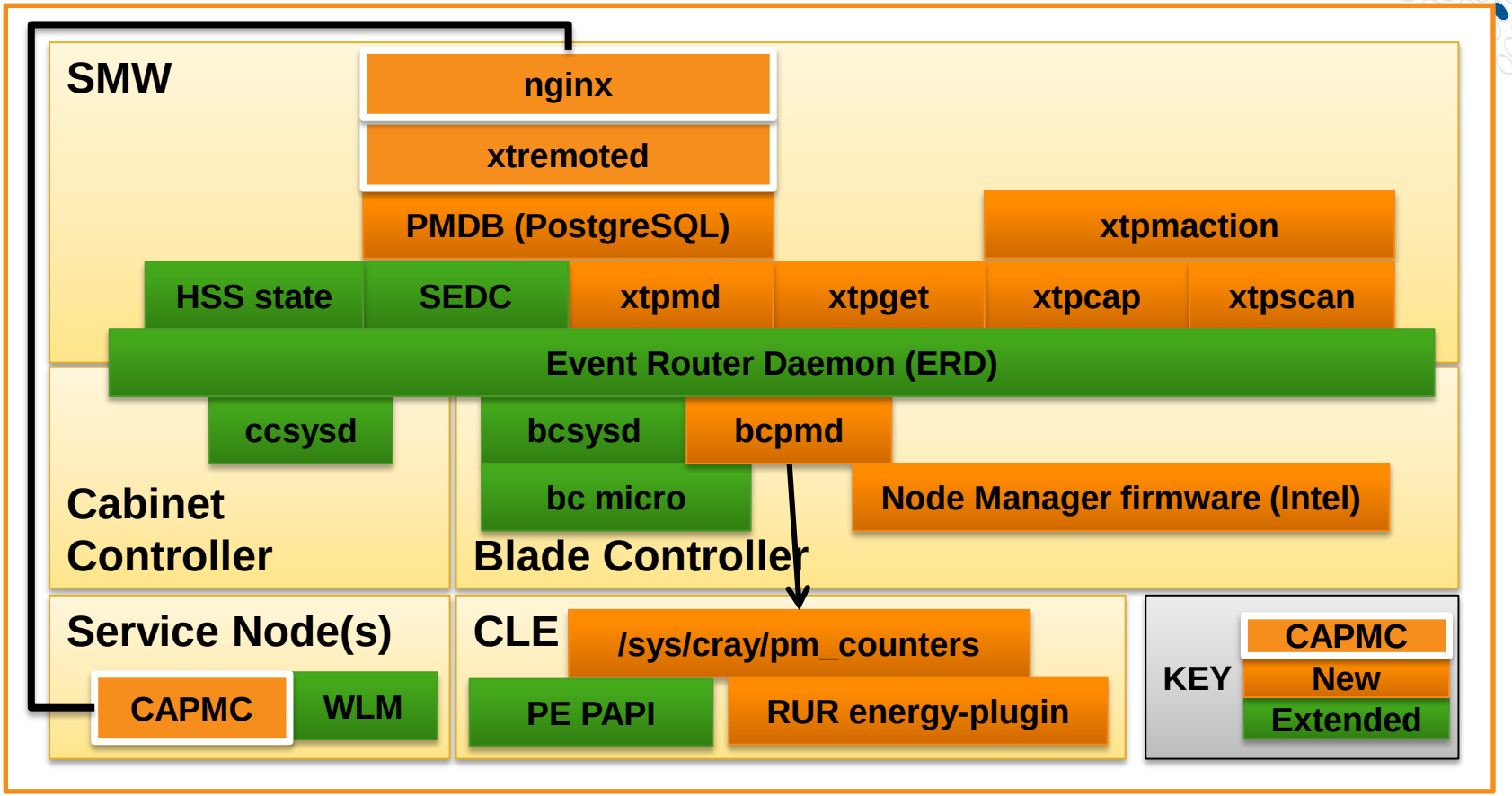
Capabilities Today on XC

Software stack
Out-of-band monitoring
In-band monitoring
Management

XC Out-Of-Band Monitoring

- **System Environmental Data Collection (SEDC)**
  - Cabinet-, blade-, and component-level data
  - Voltage, current, temperature, pressure, fan-speed, …
  - Readings updated once per minute
- **High-speed power/energy data collection**
  - Cabinet, Blade, Node, and [Accelerator] data
  - Blade level data collection at 10 Hz
- **Power Management Database (PMDB)**
  - Cabinet-level Power (+blowers)
  - Blade-, and Node-level data at 1 Hz

XC PM Software Stack



XC In-Band Monitoring

- /sys/cray/pm_counters (updated at 10Hz)

```
/sys/cray/pm_counters/accel_energy:10967358 J
/sys/cray/pm_counters/accel_power:20 W
/sys/cray/pm_counters/accel_power_cap:0 W
/sys/cray/pm_counters/energy:41006380 J
/sys/cray/pm_counters/power:55 W
/sys/cray/pm_counters/power_cap:0 W
/sys/cray/pm_counters/generation:95
/sys/cray/pm_counters/freshness:5494619
/sys/cray/pm_counters/startup:1429186804097771
/sys/cray/pm_counters/version:1
```

- Cray support for PAPI and CrayPat

XC In-Band Monitoring

- /sys/cray/pm_counters (updated at 10Hz)

```
/sys/cray/pm_counters/accel_energy:10967358 J
/sys/cray/pm_counters/accel_power:20 W
/sys/cray/pm_counters/accel_power_cap:0 W
/sys/cray/pm_counters/energy:41006380 J
/sys/cray/pm_counters/power:55 W
/sys/cray/pm_counters/power_cap:0 W
/sys/cray/pm_counters/generation:95
/sys/cray/pm_counters/freshness:5494619
/sys/cray/pm_counters/startup:1429186804097771
/sys/cray/pm_counters/version:1
```

- Cray support for PAPI and CrayPat

XC In-Band Monitoring

- /sys/cray/pm_counters (updated at 10Hz)

```
/sys/cray/pm_counters/accel_energy:10967358 J
/sys/cray/pm_counters/accel_power:20 W
/sys/cray/pm_counters/accel_power_cap:0 W
/sys/cray/pm_counters/energy:41006380 J
/sys/cray/pm_counters/power:55 W
/sys/cray/pm_counters/power_cap:0 W
/sys/cray/pm_counters/generation:95
/sys/cray/pm_counters/freshness:5494619
/sys/cray/pm_counters/startup:1429186804097771
/sys/cray/pm_counters/version:1
```

- Cray support for PAPI and CrayPat

Out-Of-Band Monitoring Use Cases

- **Real-time system monitoring with xtpget**
  - Timestamp, Current-,Average-,Peak-Power, and Accumulated Energy
  - User selectable time window for average and peak power
  - Easy command line access, no database access required
- **System-level data from PMDB**
  - System level profiling
  - Cabinet level details
  - Access days or weeks of historic data
- **Application power/energy profiling from the SMW**
  - Example text report scripts ship with the SMW release
  - Node-level power & accumulated energy data at 1 Hz
  - Application data: job-id, app-id, user, start-time, end-time, and nid-list

In-Band Monitoring Use Cases

- **Cray Resource Utilization Reporting (RUR)**
  - Application energy reporting via energy-plugin
  - Multiple reporting options
    - LLM into system logs on the SMW
    - Direct to user defined locations
    - Extendable by sites and third party workload managers
- **CrayPat & PAPI**
  - Intel RAPL counters
  - Cray custom counters
- **Direct access to /sys/cray/pm_counters**
  - Unrestricted read-only access

Control Use Cases

- **System power capping**
  - Capping >= max profiled workload
    - Avoid worst case power/cooling costs
    - Prevent budget overruns
  - More aggressive capping:
    - Can it be done while avoiding or mitigate negative performance impacts
    - Tradeoff: lower point-in-time power with increased time-to-solution
  - Capping to ride through a temporary power/cooling event

- **P-State at job launch**
  - Reduce average/peek power by running at lower frequency & voltage
  - May cause total-energy to solution to go up
  - Finding the optimum p-state

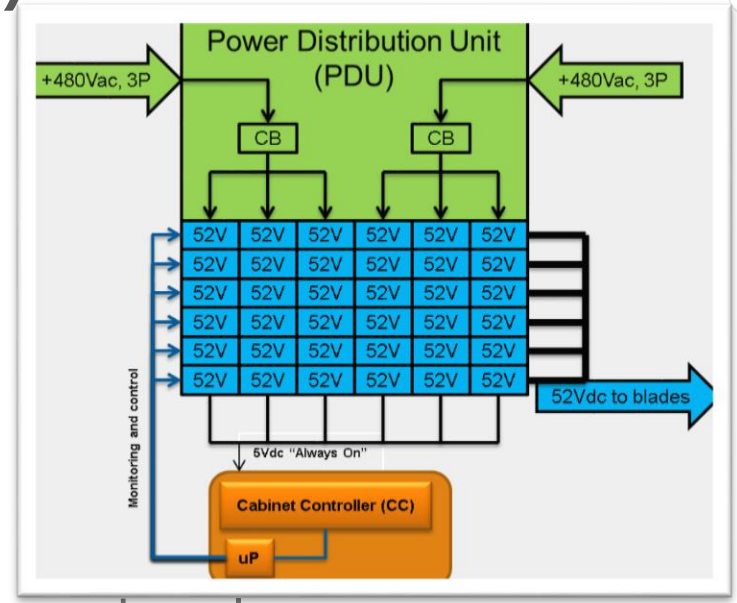
New Features released in Oct 2014

- **Cray Advanced Platform Monitoring and Control (CAPMC)**
  - Monitoring and control API for 3rd party Workload Manager integration
  - Node power: on | off, system-level and node-level monitoring
  - Enables Workload Manager (WLM) directed system-,node-, and job-level power capping
- **SEDC data → SQL database on the SMW**
  - System Environmental Data Collection (SEDC)
  - Option available (Oct 2014, SMW 7.2UP02) but not yet the default
- **Turbo-boost limiting**
  - Boot time ability to enforce max turbo boost
  - Save energy at large scale due to variation in achieved max turbo boost…

Hardware Power Monitoring

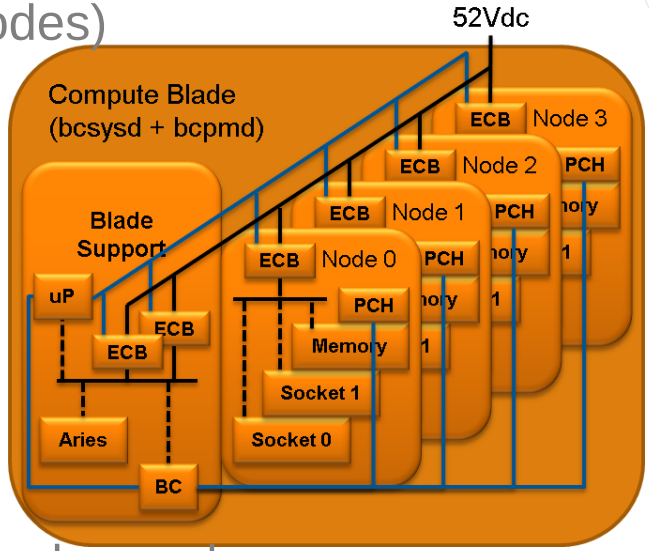
XC Cabinet Power Monitoring

- **36 bulk power supplies (rectifiers)**
  - 480 Vac to 52Vdc
  - 3000 Watts each
  - N+1 redundant
  - Hot-swap enabled
- **Cabinet controller uP**
  - Monitors rectifiers via i2c
  - Data published in BC sysfs
- **Cabinet controller software**
  - Sends power data to SMW for storages and analyses



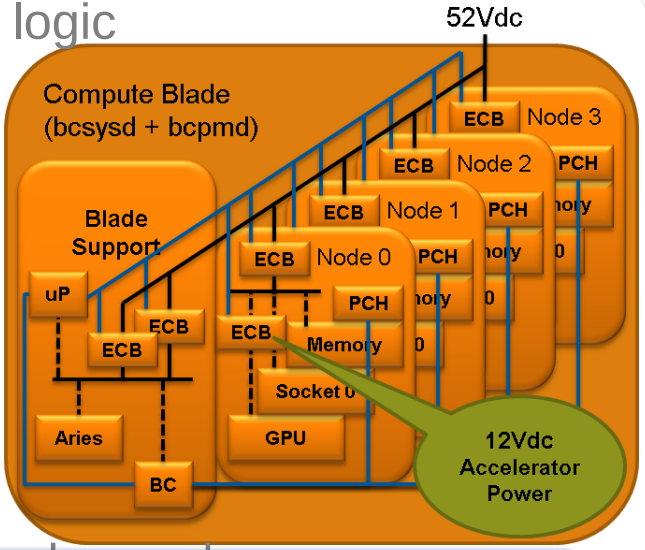
Two Socket (Xeon) Blade Power Monitoring

- **Six Electronic Circuit Breakers (ECBs)**
  - Monitor incoming 52Vdc to the blade (and nodes)
  - Two ECBs feed Aries & blade support logic
  - Four 52Vdc ECBs, one for each node
- **Blade micro-processor (uP)**
  - Monitors ECBs via i2c
  - ECBs are polled at ~= 10Hz
  - ECB data published in sysfs
- **Blade controller software**
  - Reports node power data into Intel chipset on demand
  - Sends power data to SMW for storages and analyses



Accelerated (GPU/MIC) Blade Power Monitoring

- **Ten Electronic Circuit Breakers (ECBs)**
  - Two 52Vdc ECBs feed Aries & blade support logic
  - Four 52Vdc ECB, one for each node
  - Four 12Vdc ECB, one for each accelerator
- **Blade micro-processor (uP)**
  - Monitors ECBs via i2c
  - ECBs are polled at ~= 10Hz
  - ECB data published in sysfs
- **Blade controller software**
  - Reports node power data into Intel shmem on demand
  - Sends power data to SMW for storages and analyses



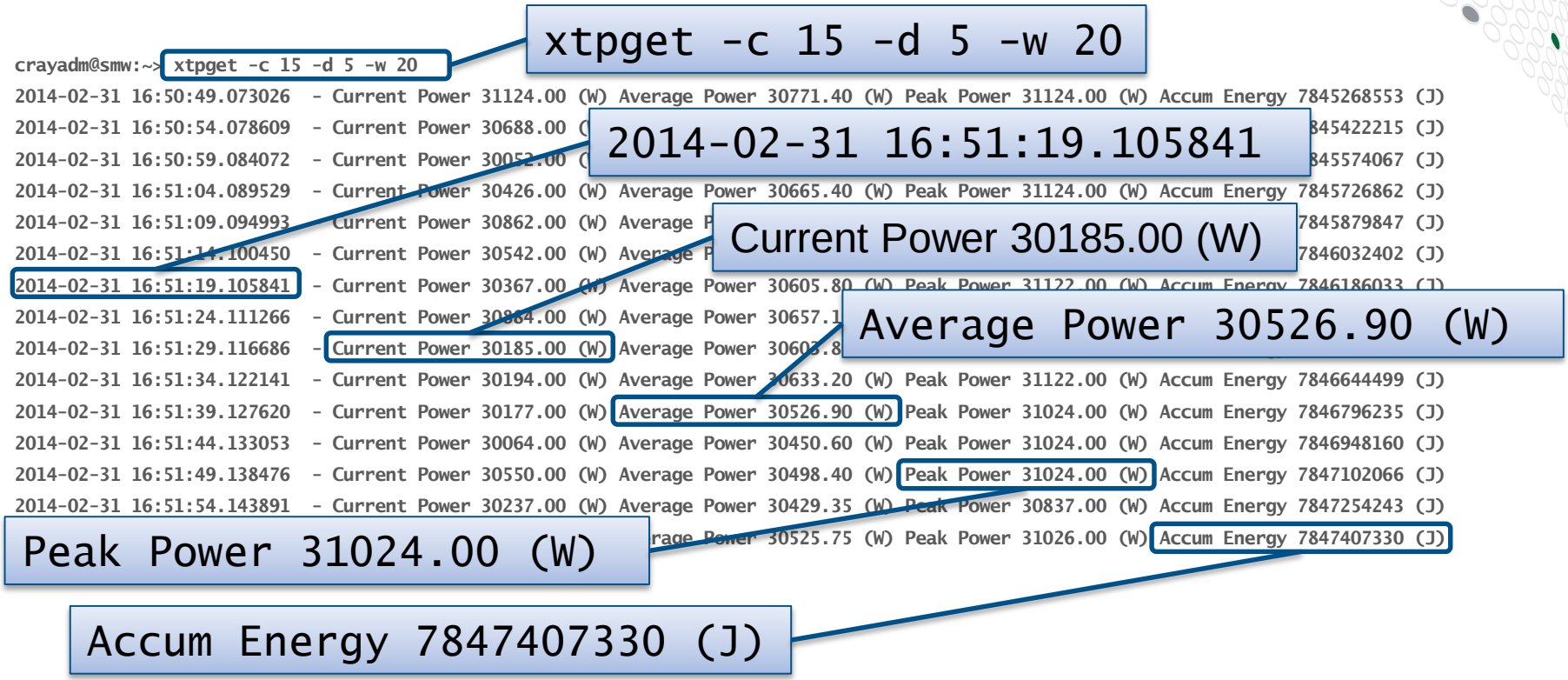
The same infrastructure supports GPU and MIC accelerated blades!

xtpget (1 of 2)

```
crayadm@smw:~> xtpget --help
Usage: xtpget [-D|--debug] [-v|--verbose] [-h|--help]
       xtpget -V|--version
       xtpget -f|--config <input file|<name=value> \
            [,<name=value>]>
       xtpget -w|--window <sample window (s)>
       xtpget -d|--delay <seconds>
       xtpget -c|--count <iterations>
```

Quick SMW command line access to total system power data

xtpget (2 of 2)



Plotting Data

Verbose Text Output

```
c0-0 CC_T_COMP_AMBIENT_TEMP0(MAX = 23.800000 MEDIAN = 23.400000 AVG = 23.297600)
c0-0 CC_T_COMP_AMBIENT_TEMP1(MAX = 22.800000 MEDIAN = 22.100000 AVG = 22.120000)
c0-1 CC_T_COMP_AMBIENT_TEMP0(MAX = 22.300000 MEDIAN = 21.500000 AVG = 21.555200)
…
c9-1 CC_T_COMP_WATER_TEMP_OUT(MAX = 22.400000 MEDIAN = 19.800000 AVG = 19.830400)
System DC Power (MAX = 1541254 , AVG = 328265 , MEDIAN = 303892) Watts
    c0-0 CAB_DC_POWER(MAX = 74210 MEDIAN = 12519 AVG = 13644) Watts
    c0-1 CAB_DC_POWER(MAX = 74307 MEDIAN = 12336 AVG = 13523) Watts
    c1-0 CAB_DC_POWER(MAX = 73331 MEDIAN = 12465 AVG = 13617) Watts
    …
    c9-1 CAB_DC_POWER(MAX = 74563 MEDIAN = 12367 AVG = 13582) Watts
```

Cabinet

Verbose Text Output

```
c0-0 CC_T_COMP_AMBIENT_TEMP0(MAX = 23.800000 MEDIAN = 23.400000 AVG = 23.297600)
c0-0 CC_T_COMP_AMBIENT_TEMP1(MAX = 22.800000 MEDIAN = 22.100000 AVG = 22.120000)
c0-1 CC_T_COMP_AMBIENT_TEMP0(MAX = 22.300000 MEDIAN = 21.500000 AVG = 21.555200)
…
c9-1 CC_T_COMP_WATER_TEMP_OUT(MAX = 22.400000 MEDIAN = 19.800000 AVG = 19.830400)
System DC Power (MAX = 1541254 , AVG = 328265 , MEDIAN = 303892) Watts
    c0-0 CAB_DC_POWER(MAX = 74210 MEDIAN = 12519 AVG = 13644) Watts
    c0-1 CAB_DC_POWER(MAX = 74307 MEDIAN = 12336 AVG = 13523) Watts
    c1-0 CAB_DC_POWER(MAX = 73331 MEDIAN = 12465 AVG = 13617) Watts
    …
    c9-1 CAB_DC_POWER(MAX = 74563 MEDIAN = 12367 AVG = 13582) Watts
```

Sensor Name

Verbose Text Output

```
c0-0 CC_T_COMP_AMBIENT_TEMP0(MAX = 23.800000 MEDIAN = 23.400000 AVG = 23.297600)
c0-0 CC_T_COMP_AMBIENT_TEMP1(MAX = 22.800000 MEDIAN = 22.100000 AVG = 22.120000)
c0-1 CC_T_COMP_AMBIENT_TEMP0(MAX = 22.300000 MEDIAN = 21.500000 AVG = 21.555200)
…
c9-1 CC_T_COMP_WATER_TEMP_OUT(MAX = 22.400000 MEDIAN = 19.800000 AVG = 19.830400)
System DC Power (MAX = 1541254 , AVG = 328265 , MEDIAN = 303892) Watts
    c0-0 CAB_DC_POWER(MAX = 74210 MEDIAN = 12519 AVG = 13644) Watts
    c0-1 CAB_DC_POWER(MAX = 74307 MEDIAN = 12336 AVG = 13523) Watts
    c1-0 CAB_DC_POWER(MAX = 73331 MEDIAN = 12465 AVG = 13617) Watts
    …
    c9-1 CAB_DC_POWER(MAX = 74563 MEDIAN = 12367 AVG = 13582) Watts
```

Maximum, Median, Average sensor readings for the time window data was collection

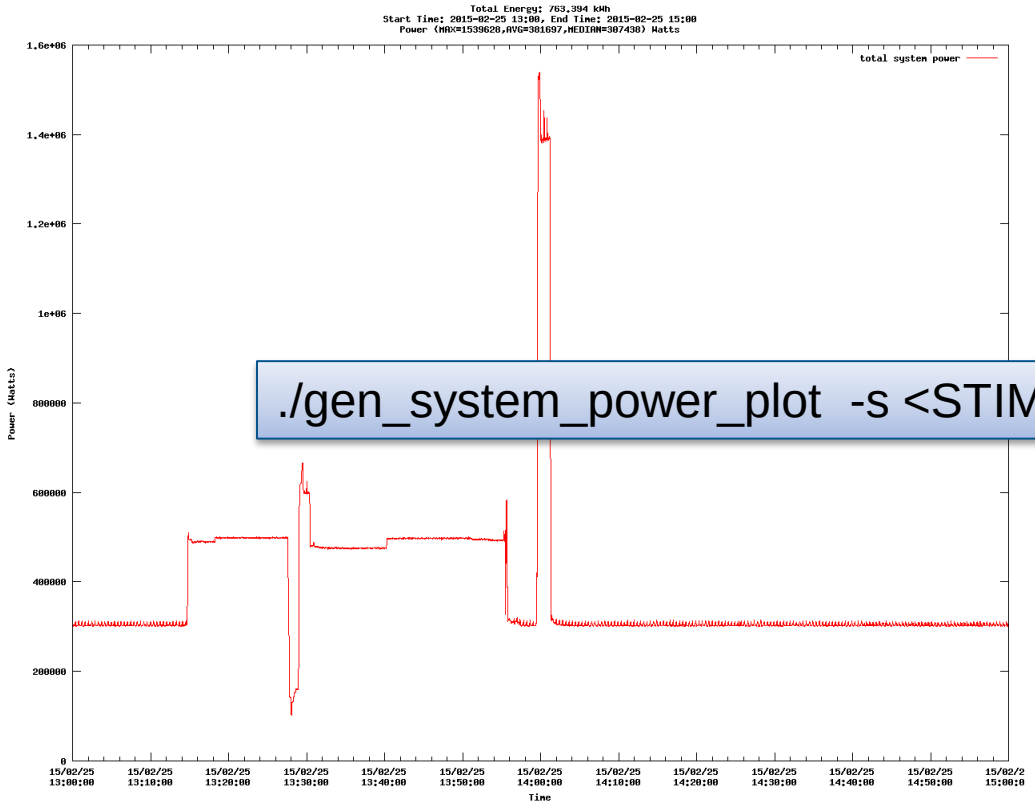
Verbose Text Output

```
c0-0 CC_T_COMP_AMBIENT_TEMP0(MAX = 23.800000 MEDIAN = 23.400000 AVG = 23.297600)
c0-0 CC_T_COMP_AMBIENT_TEMP1(MAX = 22.800000 MEDIAN = 22.100000 AVG = 22.120000)
c0-1 CC_T_COMP_AMBIENT_TEMP0(MAX = 22.300000 MEDIAN = 21.500000 AVG = 21.555200)
…
c9-1 CC_T_COMP_WATER_TEMP_OUT(MAX = 22.400000 MEDIAN = 19.800000 AVG = 19.830400)
System DC Power (MAX = 1541254 , AVG = 328265 , MEDIAN = 303892) Watts
    c0-0 CAB_DC_POWER(MAX = 74210 MEDIAN = 12519 AVG = 13644) Watts
    c0-1 CAB_DC_POWER(MAX = 74307 MEDIAN = 12336 AVG = 13523) Watts
    c1-0 CAB_DC_POWER(MAX = 73331 MEDIAN = 12465 AVG = 13617) Watts
    …
    c9-1 CAB_DC_POWER(MAX = 74563 MEDIAN = 12367 AVG = 13582) Watts
```

Maximum, Median, Average: System power (Cabinet DC + Blower Power)

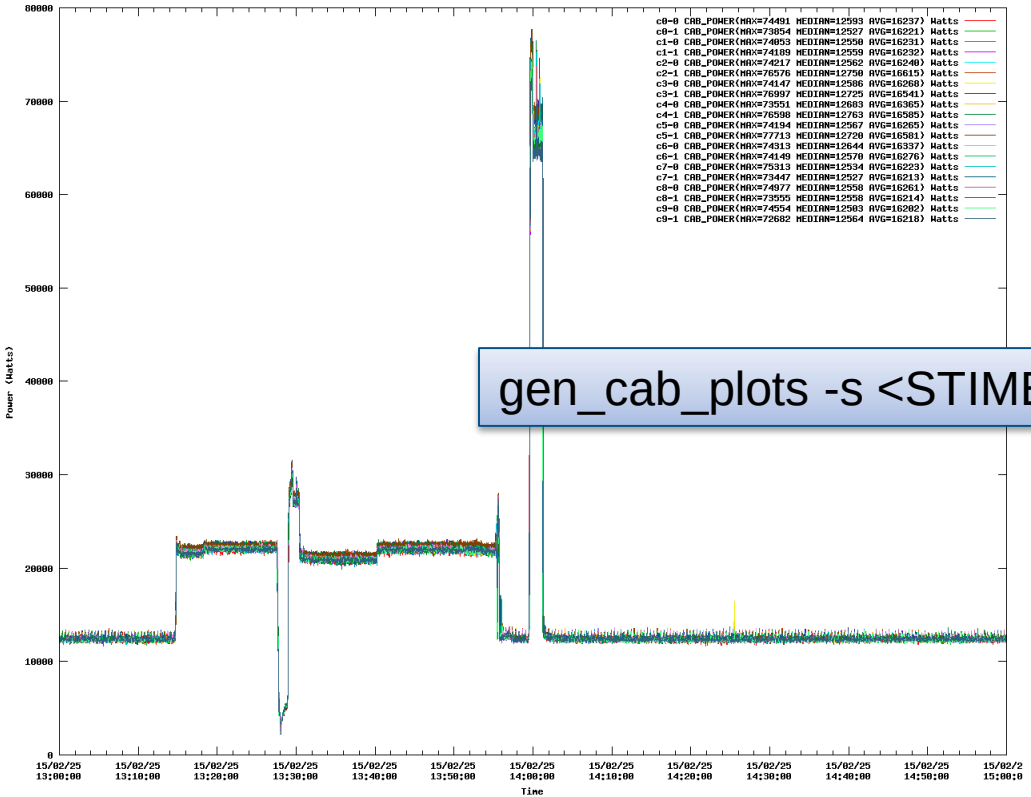
- Needs to be scaled to estimate AC power

System Power DC, 20 Cabinets Including Blowers



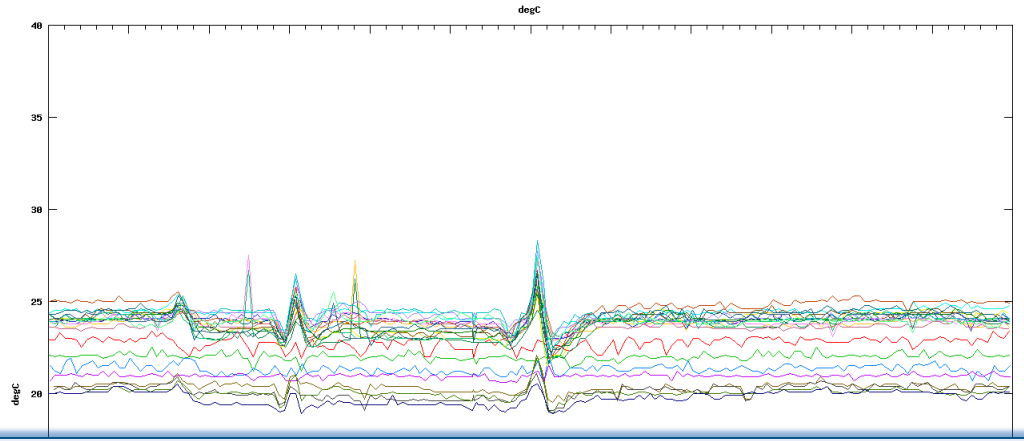
./gen_system_power_plot -s <STIME> -e <ETIME>

Cabinet Power (20 Cabinets)



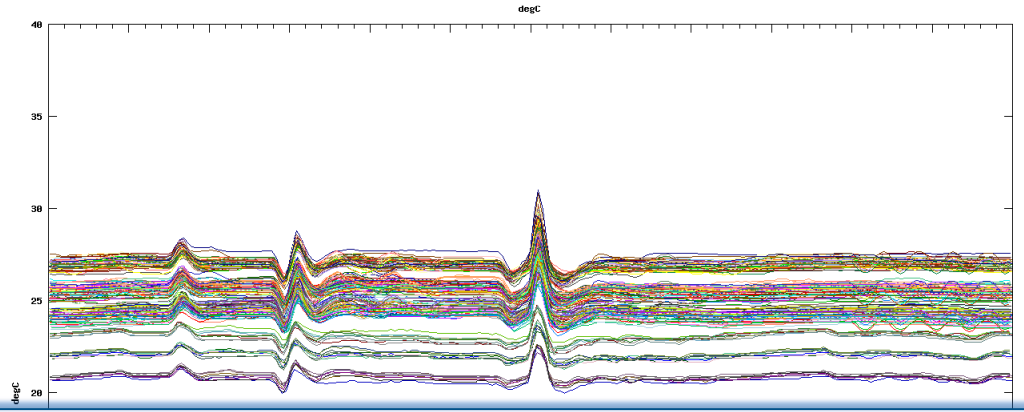
gen_cab_plots -s <STIME> -e <ETIME>

Blower Cab Ambient Temp



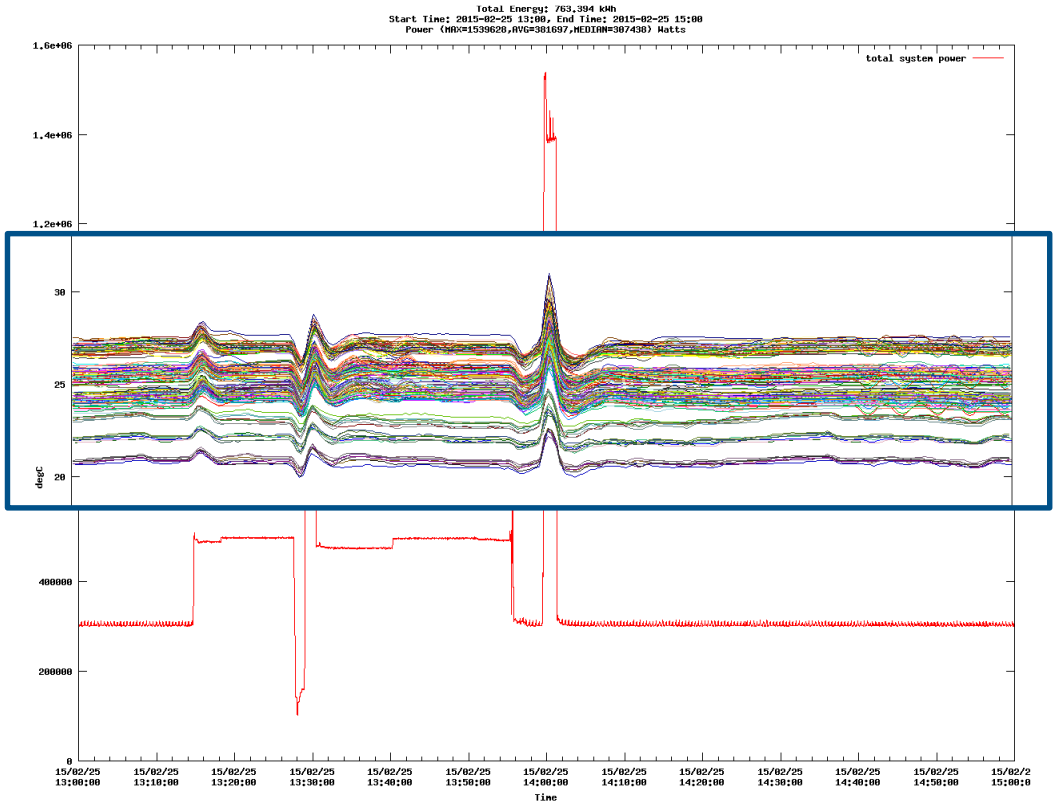
- Blower Cab Ambient Temp:
  - CC_T_COMP_AMBIENT_TEMP[0-1]
  - 2 sensors / blower-cabinet * 12 blower-cabinets
- BC_AIR_TEMP="1006,1007"
- PMDB_DB="pmdb.cc_sedc_data" ./gen_sedc_plots -Zvvi $BC_AIR_TEMP \
  -s <STIME> -e <ETIME>

CC Air Stream Temp



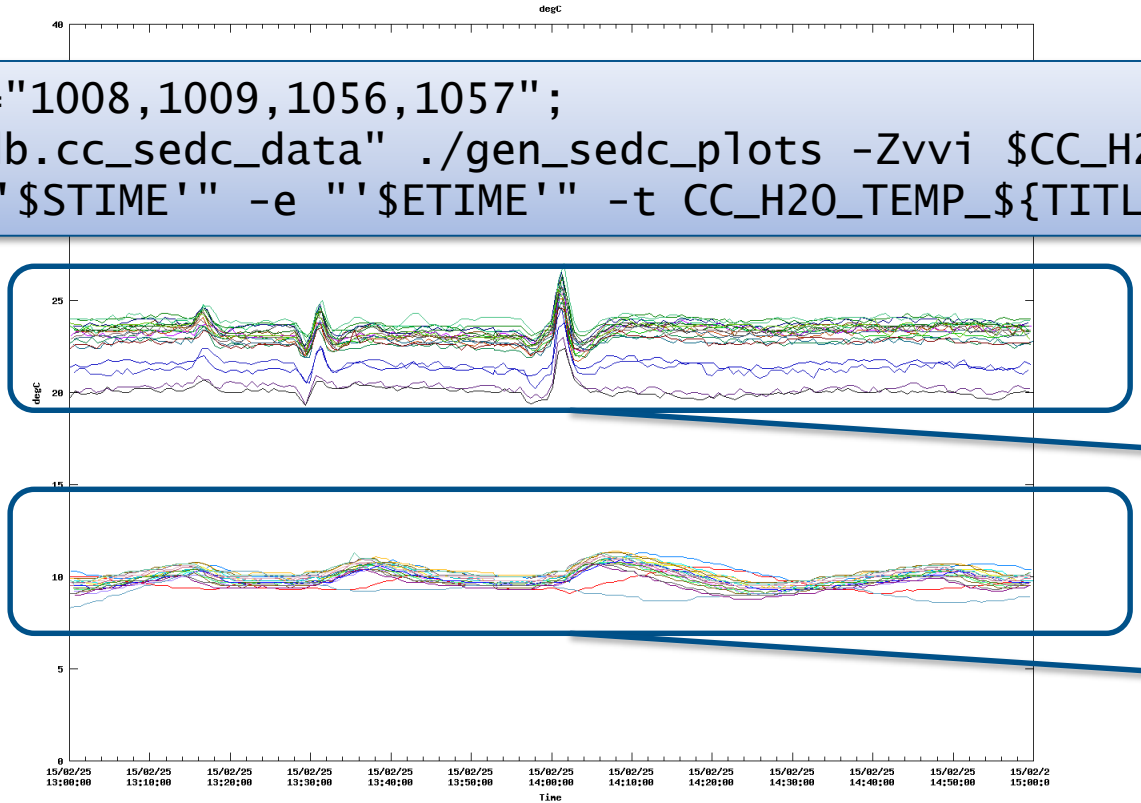
- CC Air Stream Temp:
  - CC_T_COMP_CH[0-2]_AIR_TEMP[0-1]
  - 12 sensors / cabinet * 20 Cabinets
- CC_AIR_TEMP="1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021";
- PMDB_DB="pmdb.cc_sedc_data" ./gen_sedc_plots -Zvvi $CC_AIR_TEMP \
  -s <STIME> -e <ETIME>

CC Air Stream Temp & System Power Plots



Water Temp In & Out

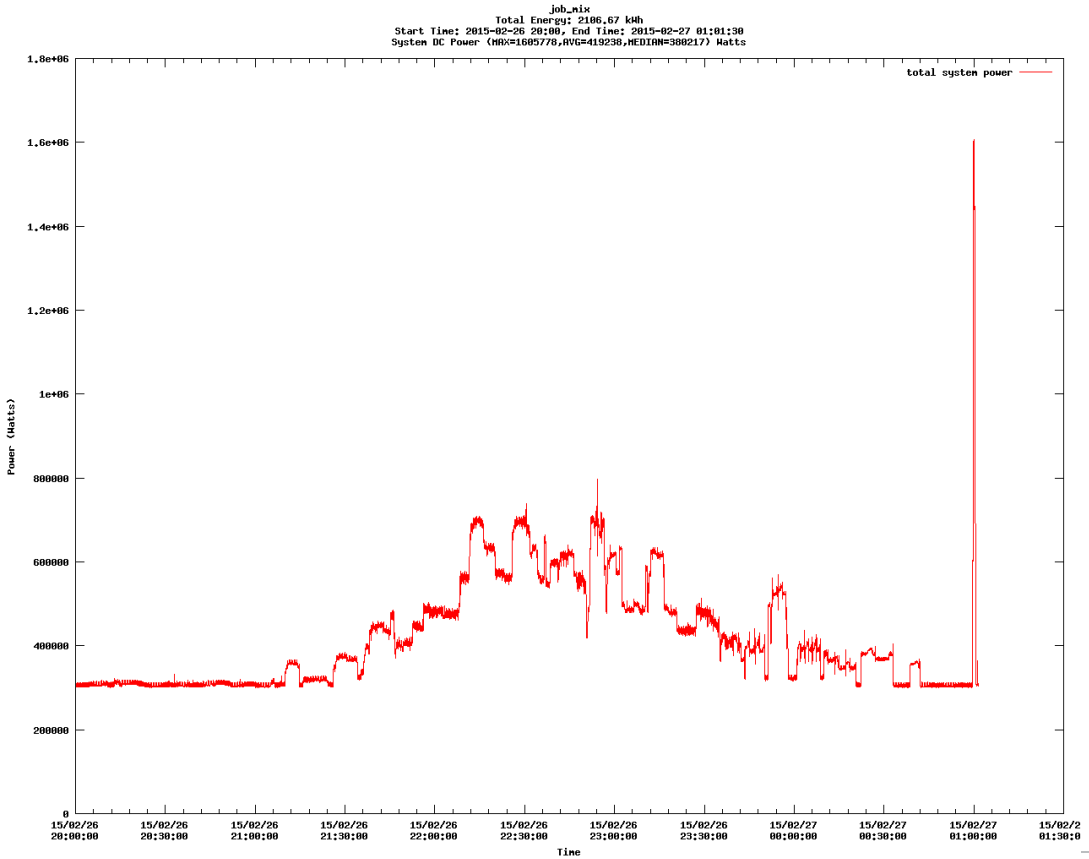
```
CC_H2O_TEMP="1008,1009,1056,1057";
PMDB_DB="pmdb.cc_sedc_data" ./gen_sedc_plots -Zvvi $CC_H2O_TEMP \
        -s "'$STIME'" -e "'$ETIME'" -t CC_H2O_TEMP_${TITLE} -z
```



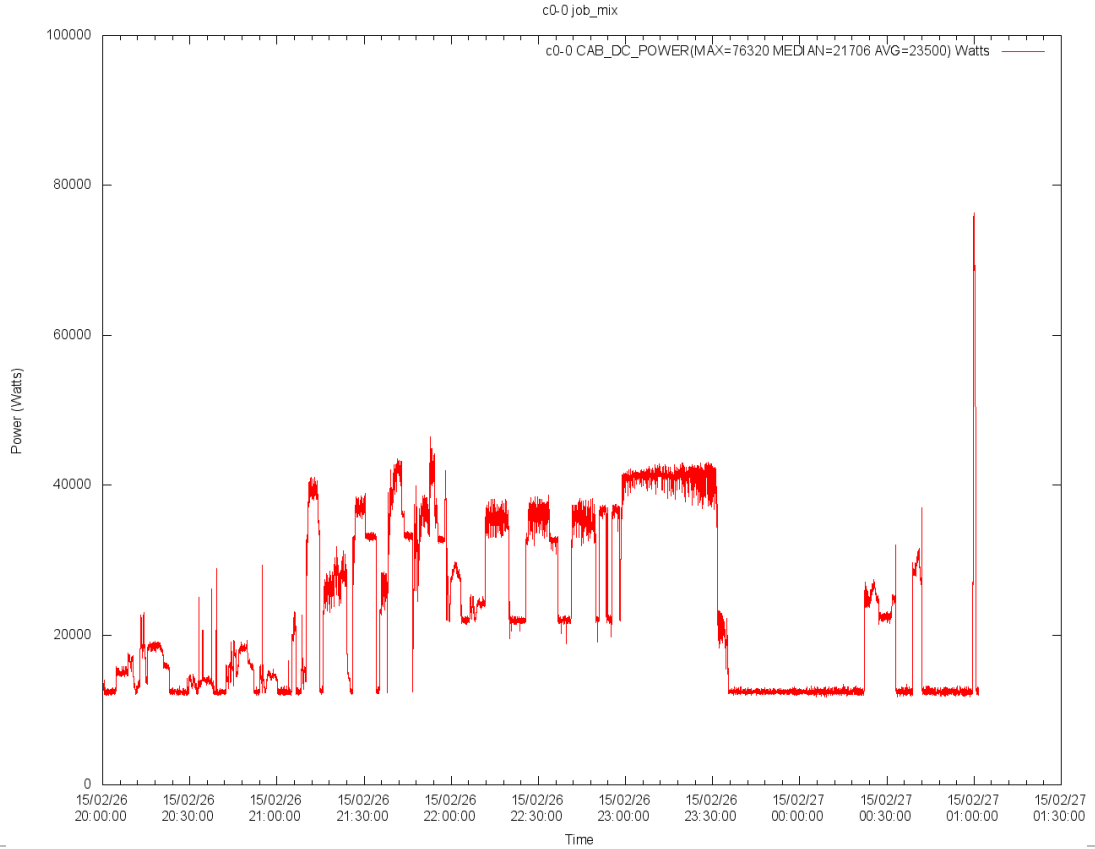
Outlet Water

Inlet Water

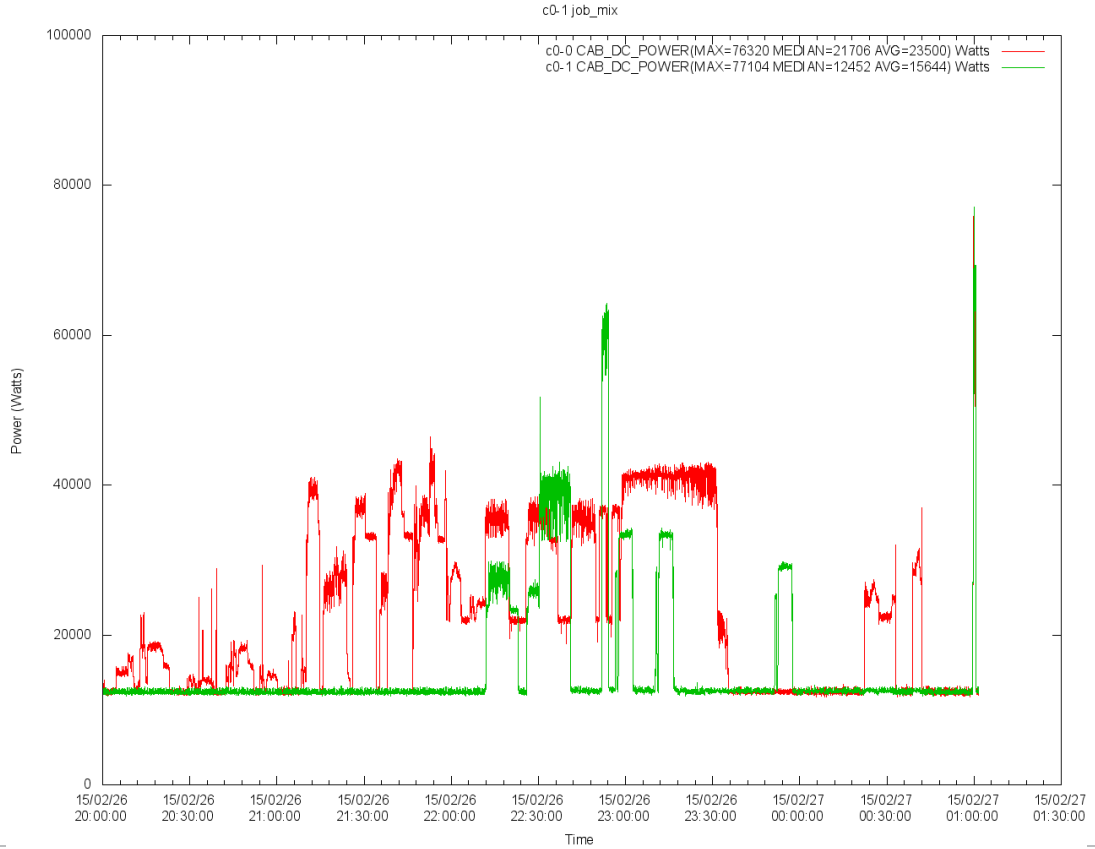
Job Mix: System Power



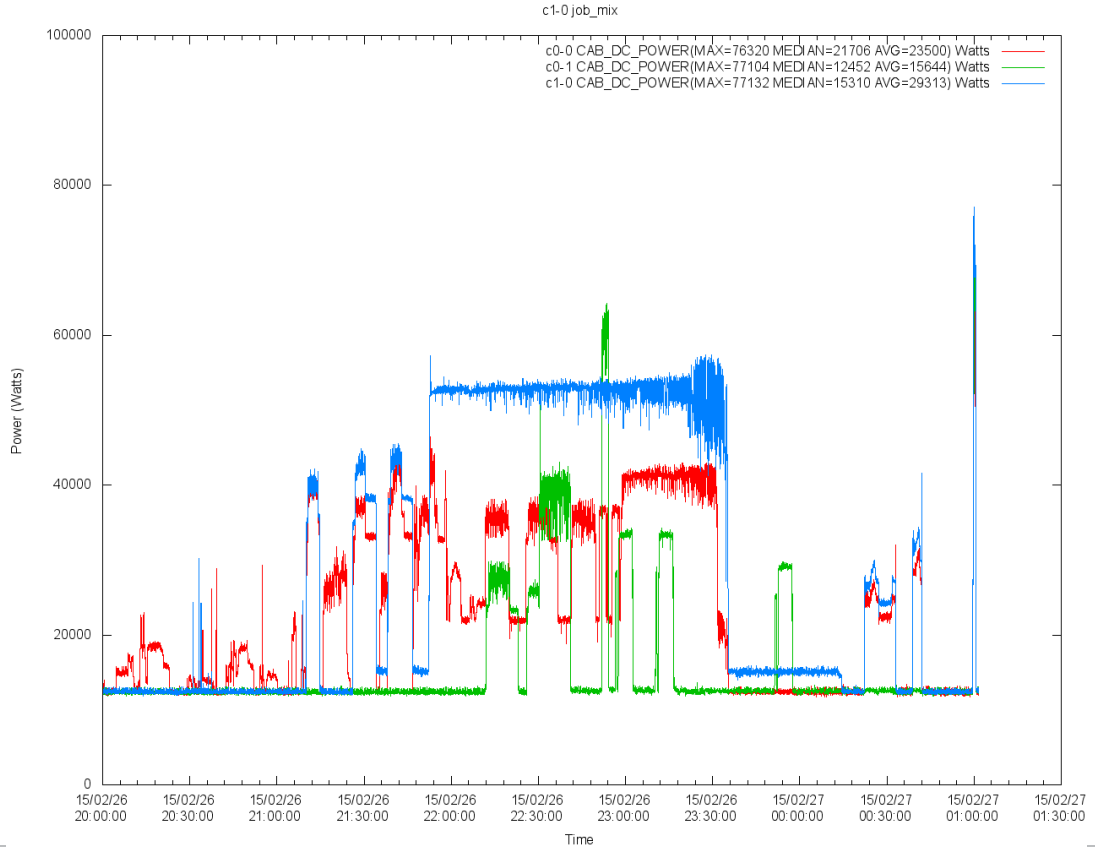
Job Mix: Cabinet Power



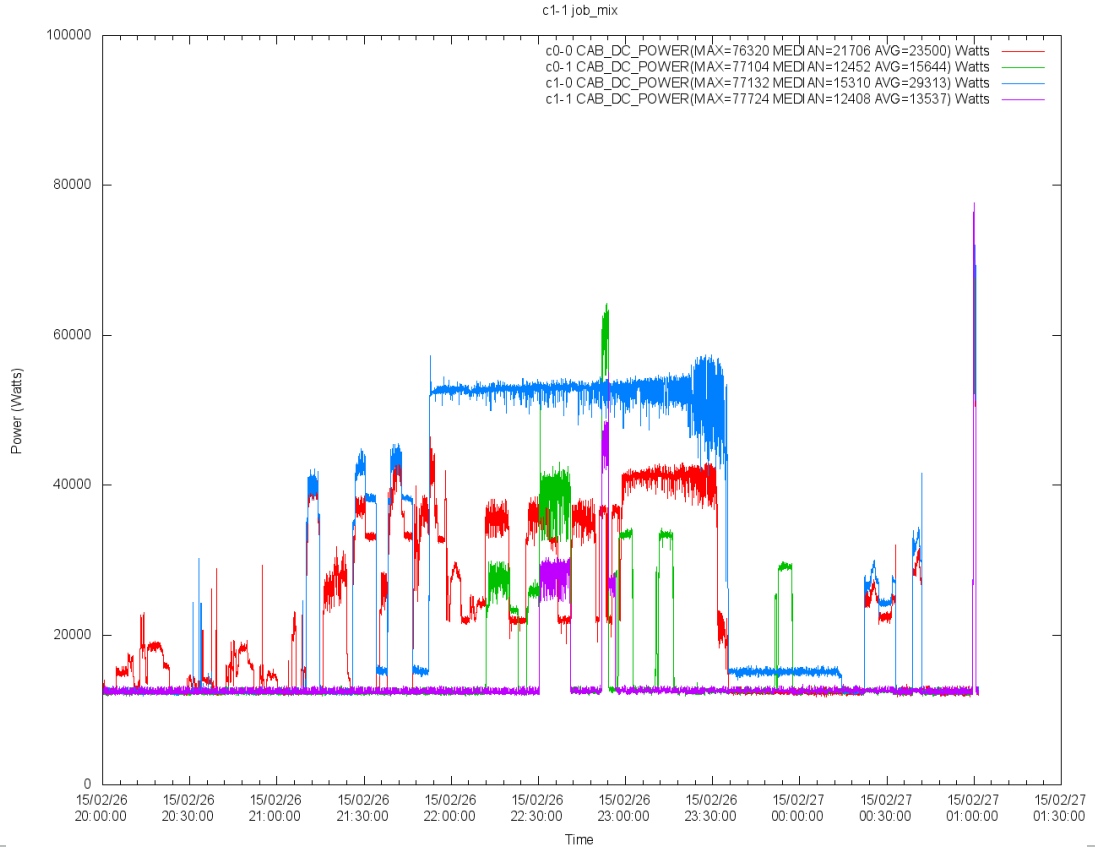
Job Mix: Cabinet Power



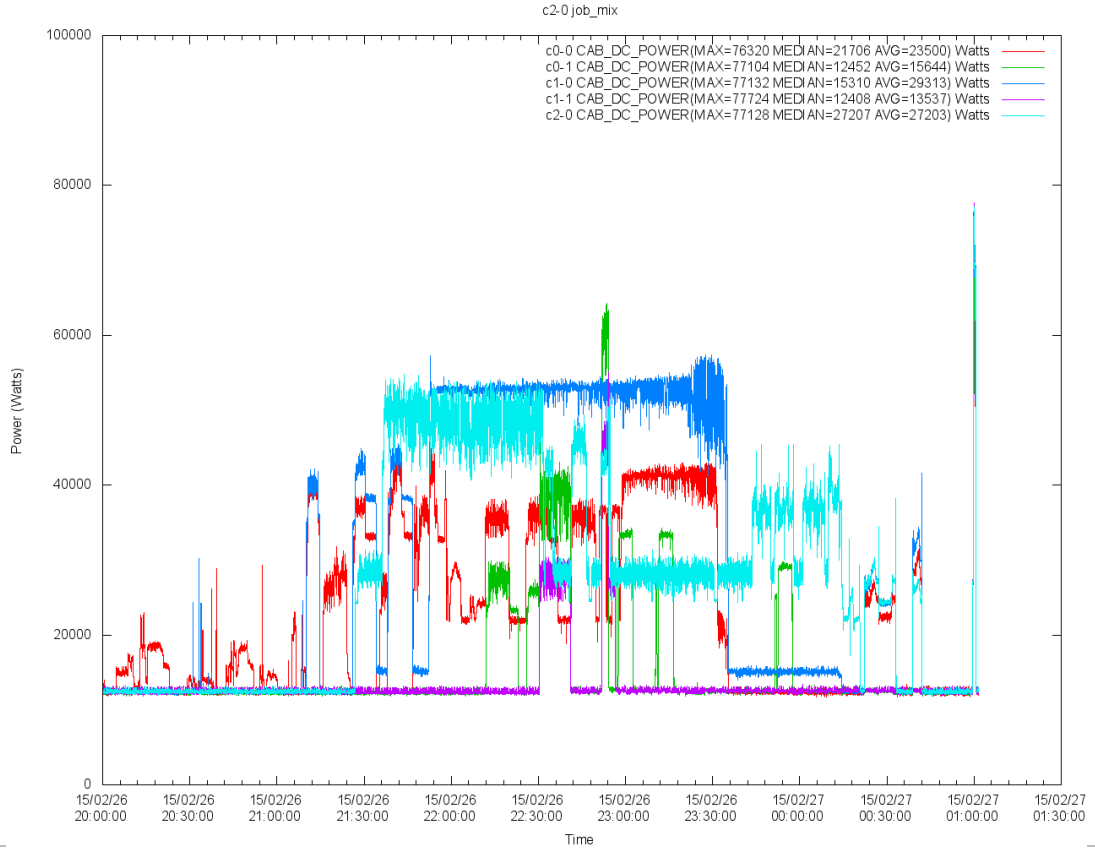
Job Mix: Cabinet Power



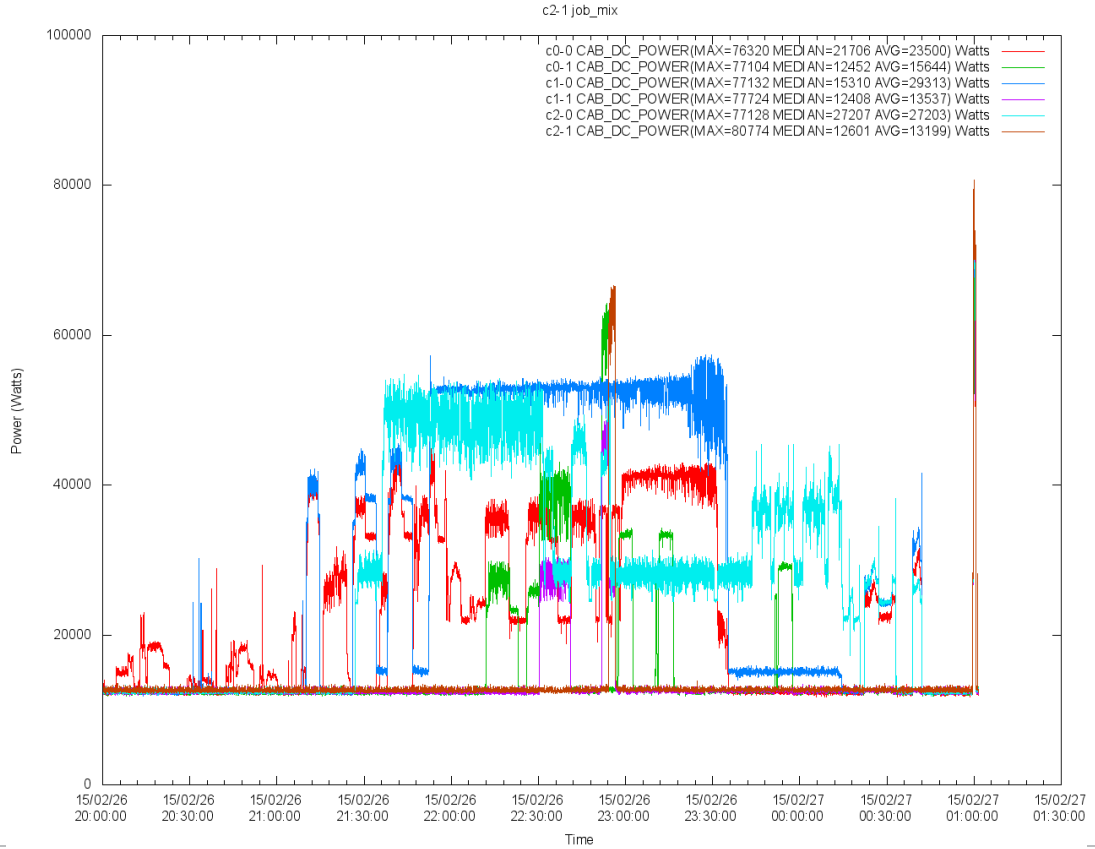
Job Mix: Cabinet Power



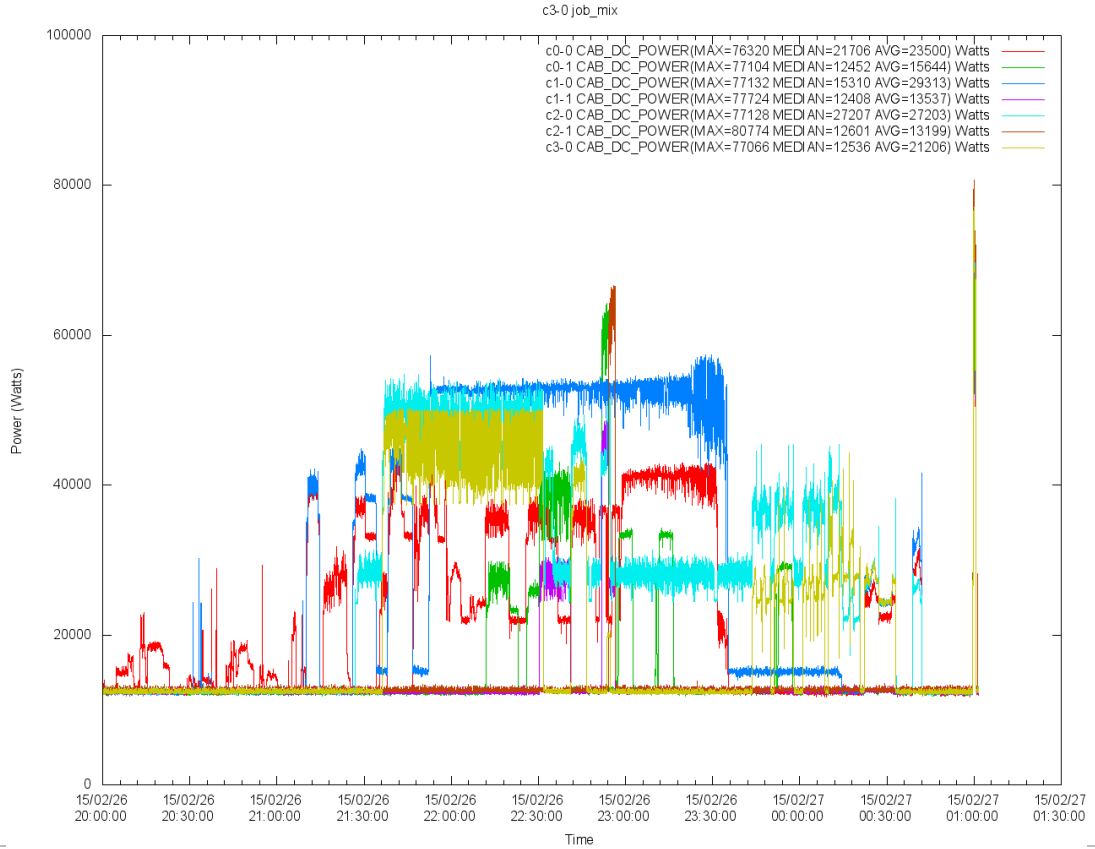
Job Mix: Cabinet Power



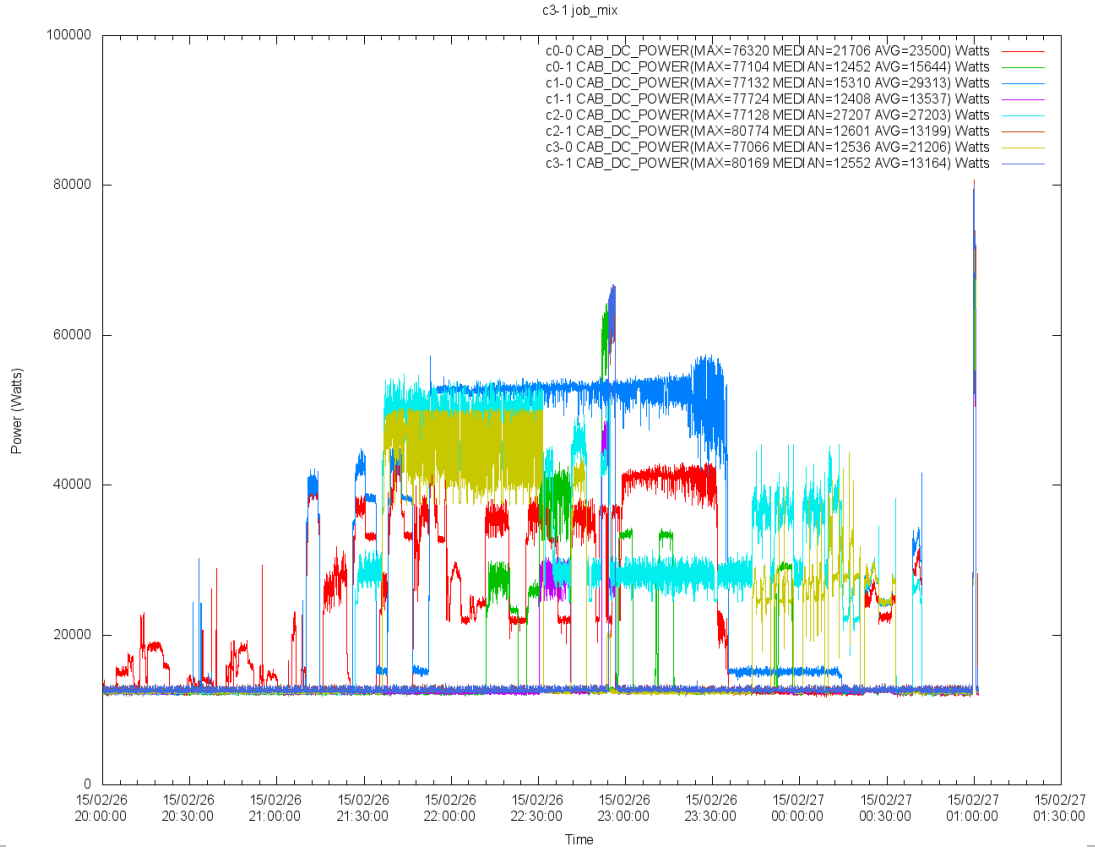
Job Mix: Cabinet Power



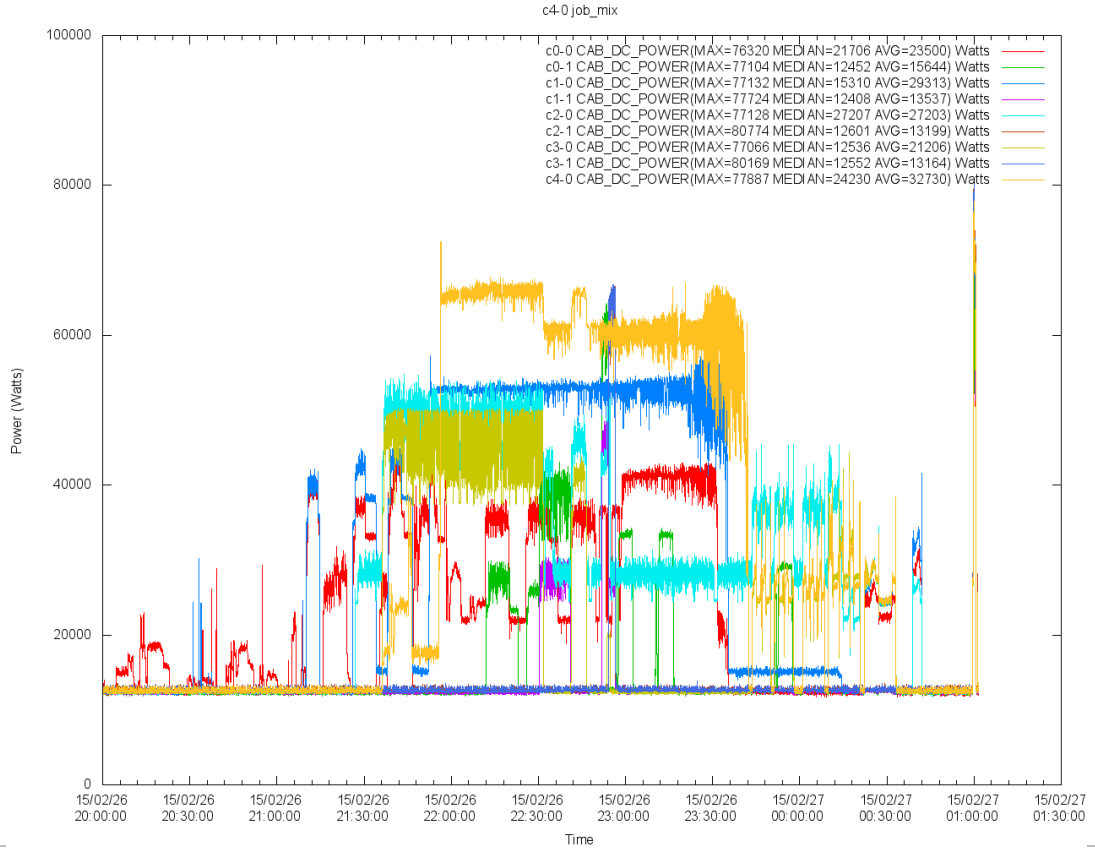
Job Mix: Cabinet Power



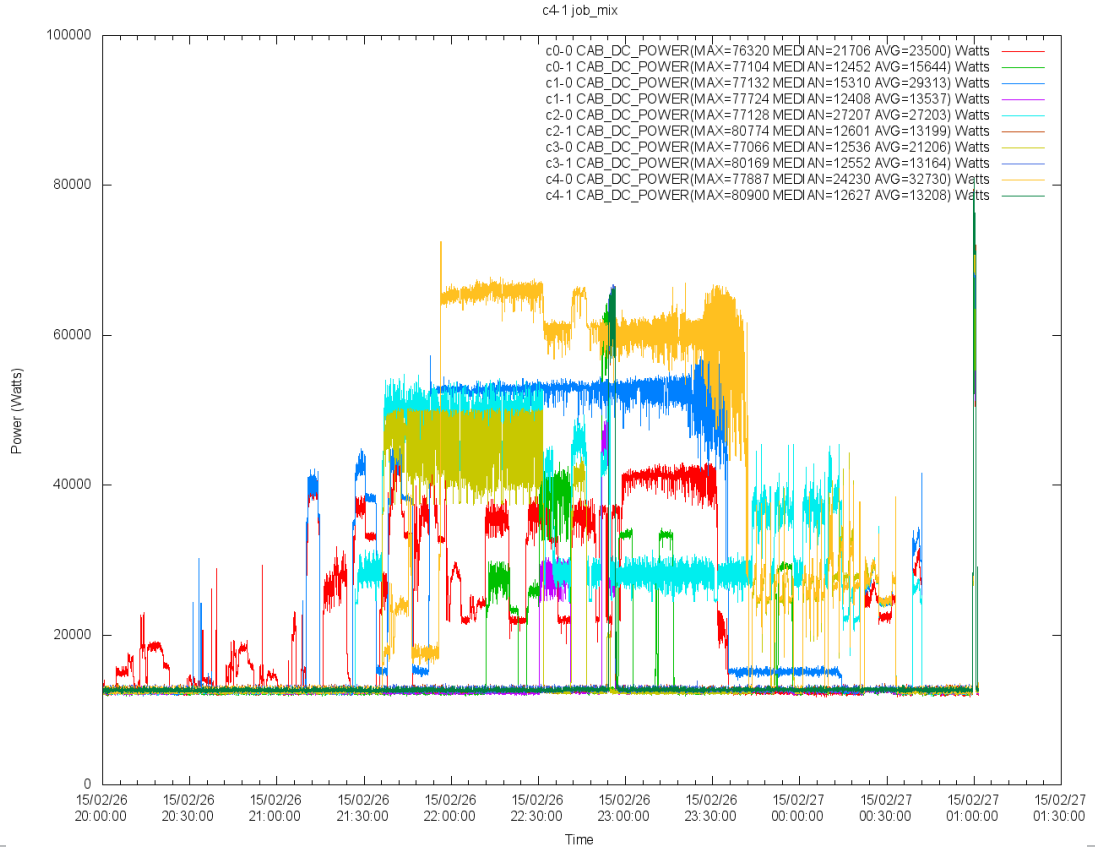
Job Mix: Cabinet Power



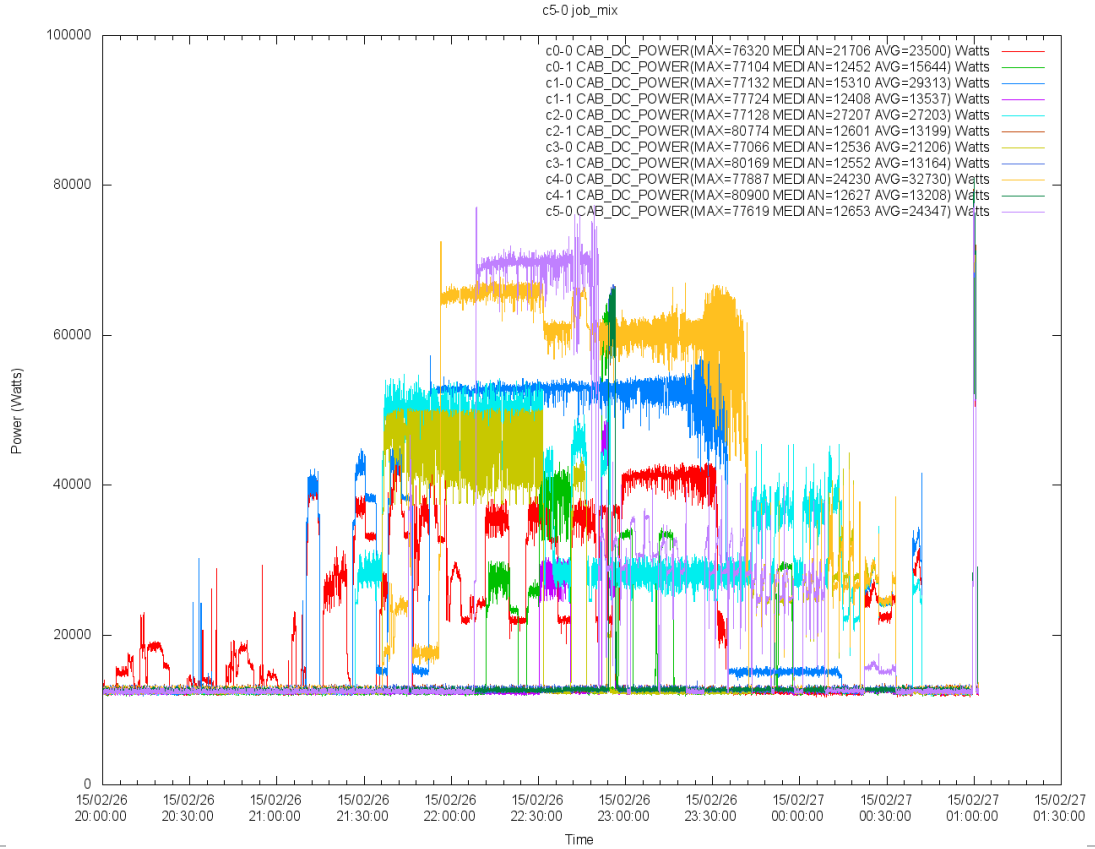
Job Mix: Cabinet Power



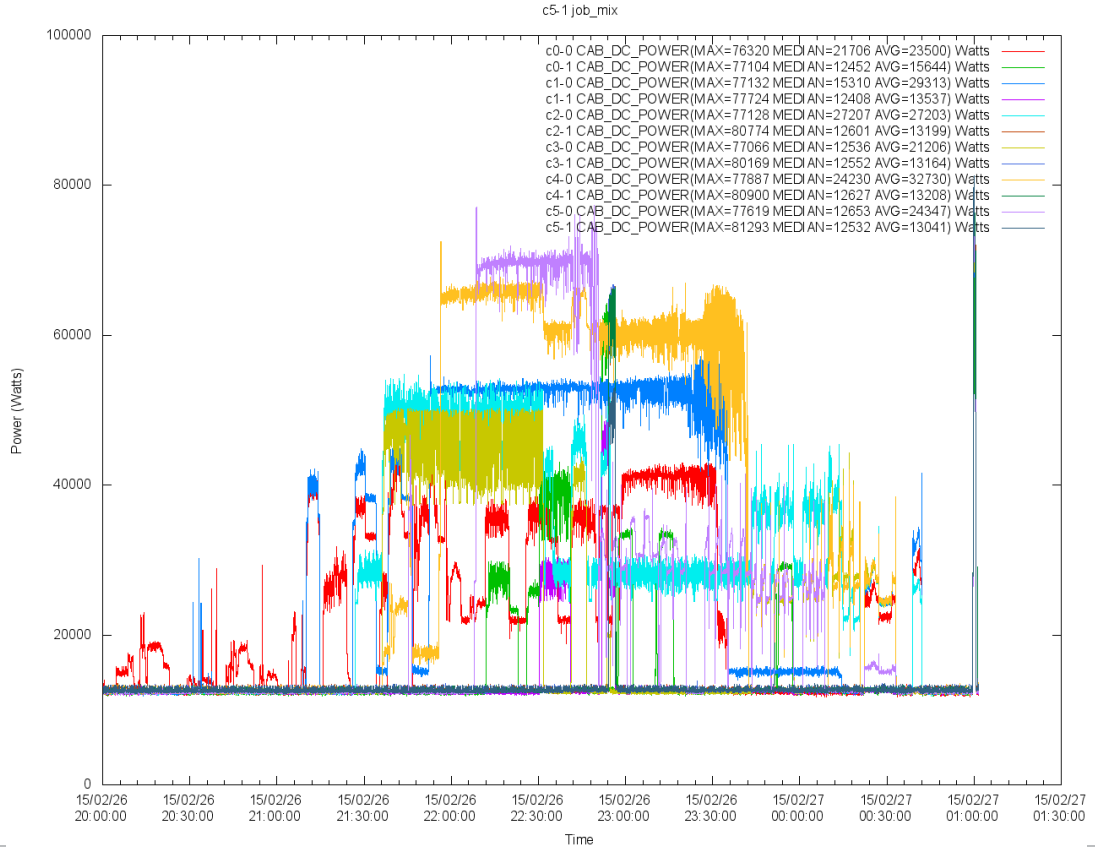
Job Mix: Cabinet Power



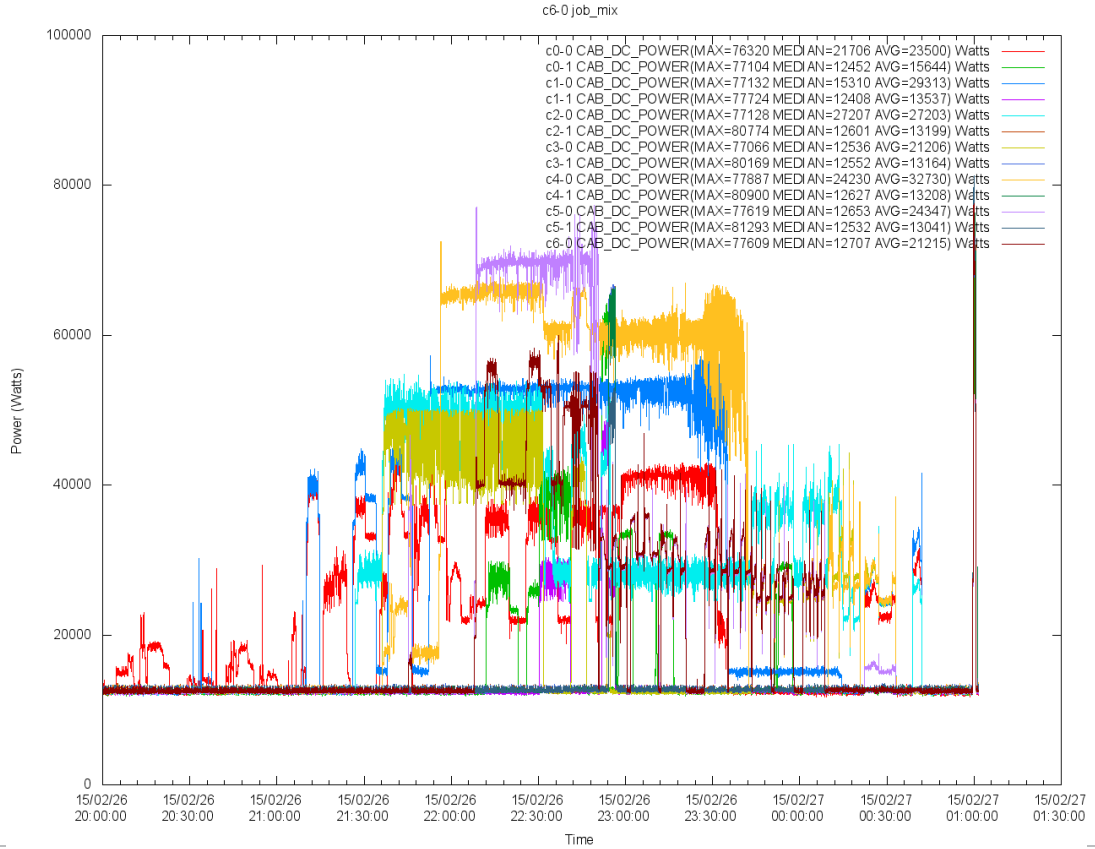
Job Mix: Cabinet Power



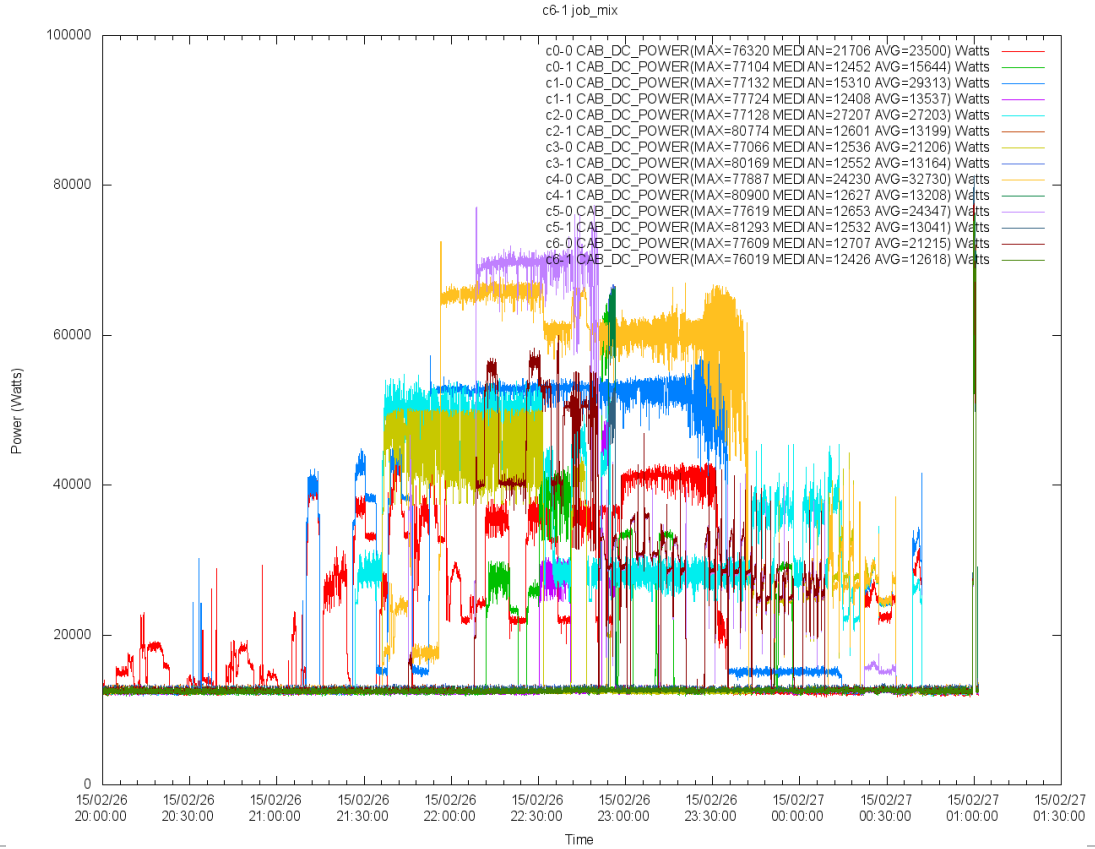
Job Mix: Cabinet Power



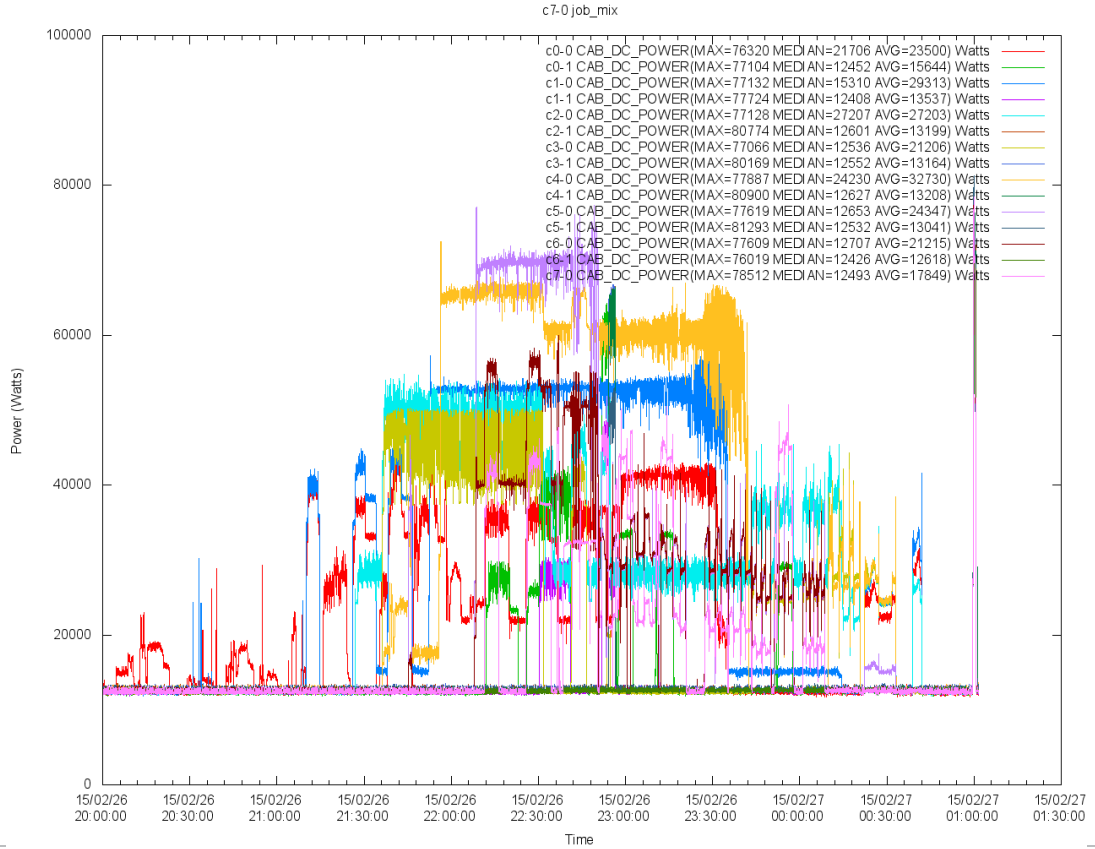
Job Mix: Cabinet Power



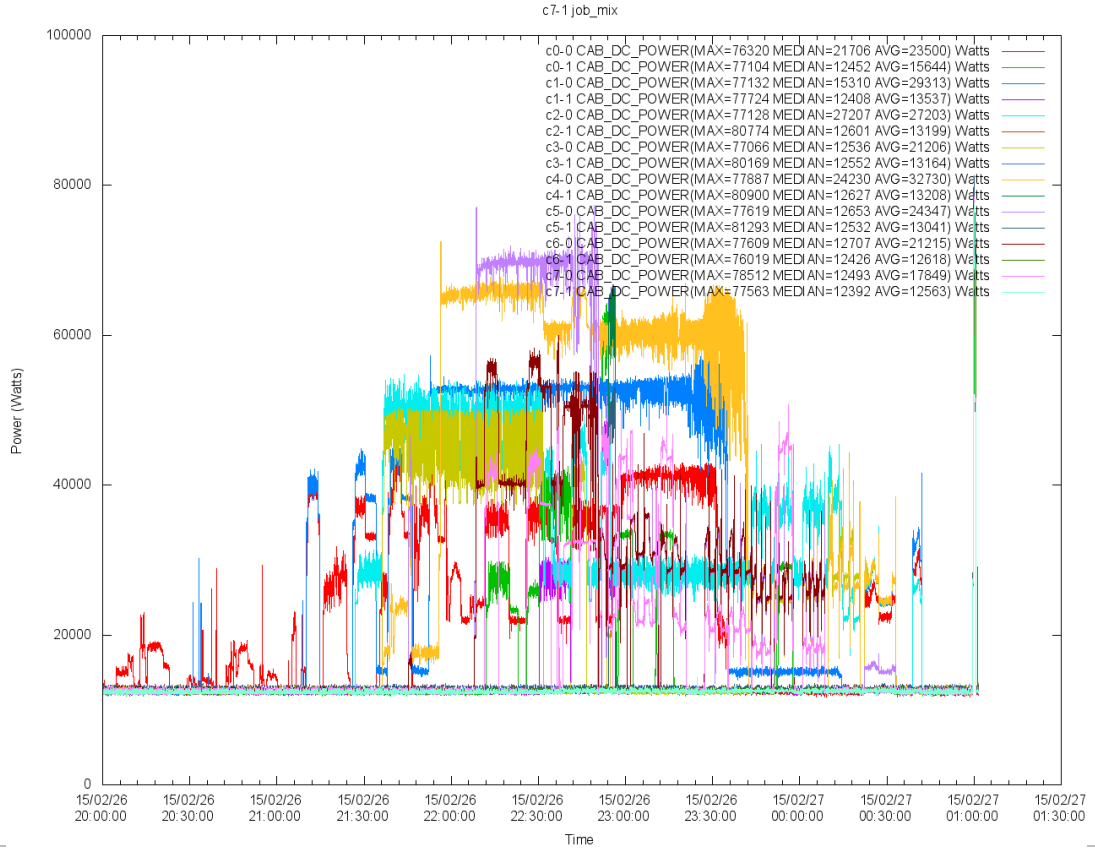
Job Mix: Cabinet Power



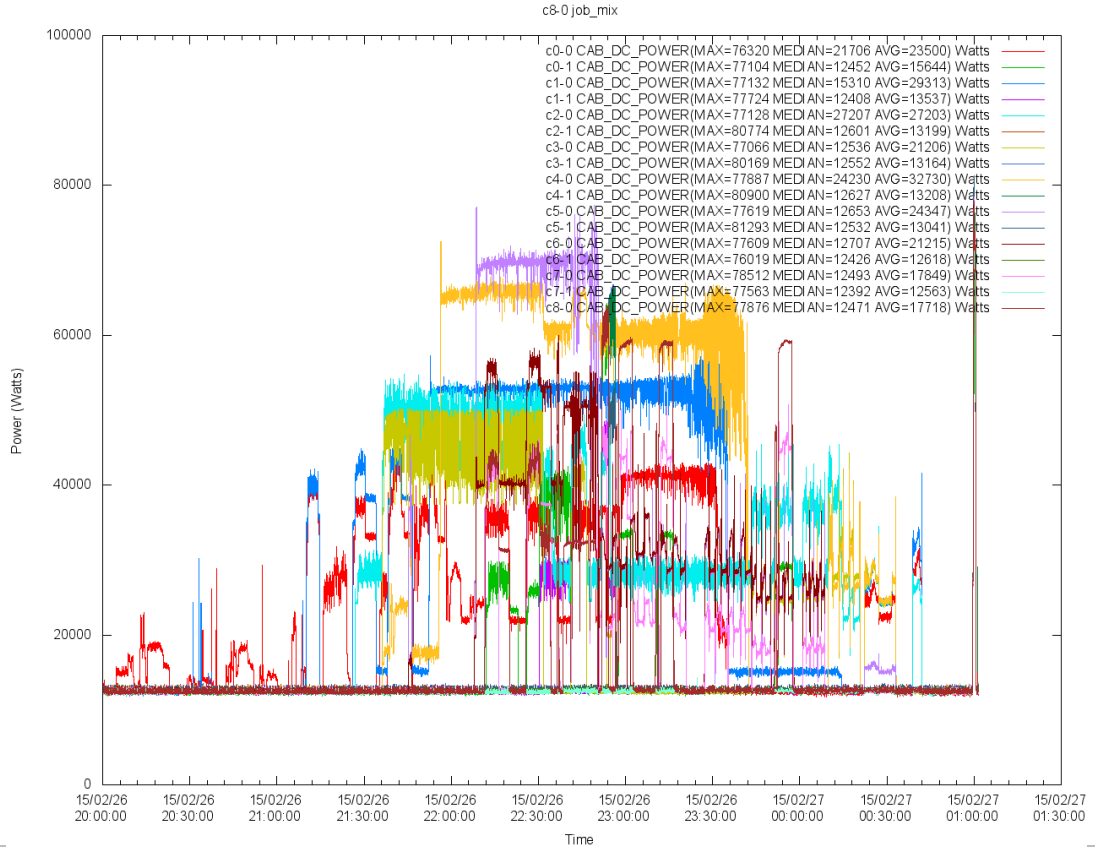
Job Mix: Cabinet Power



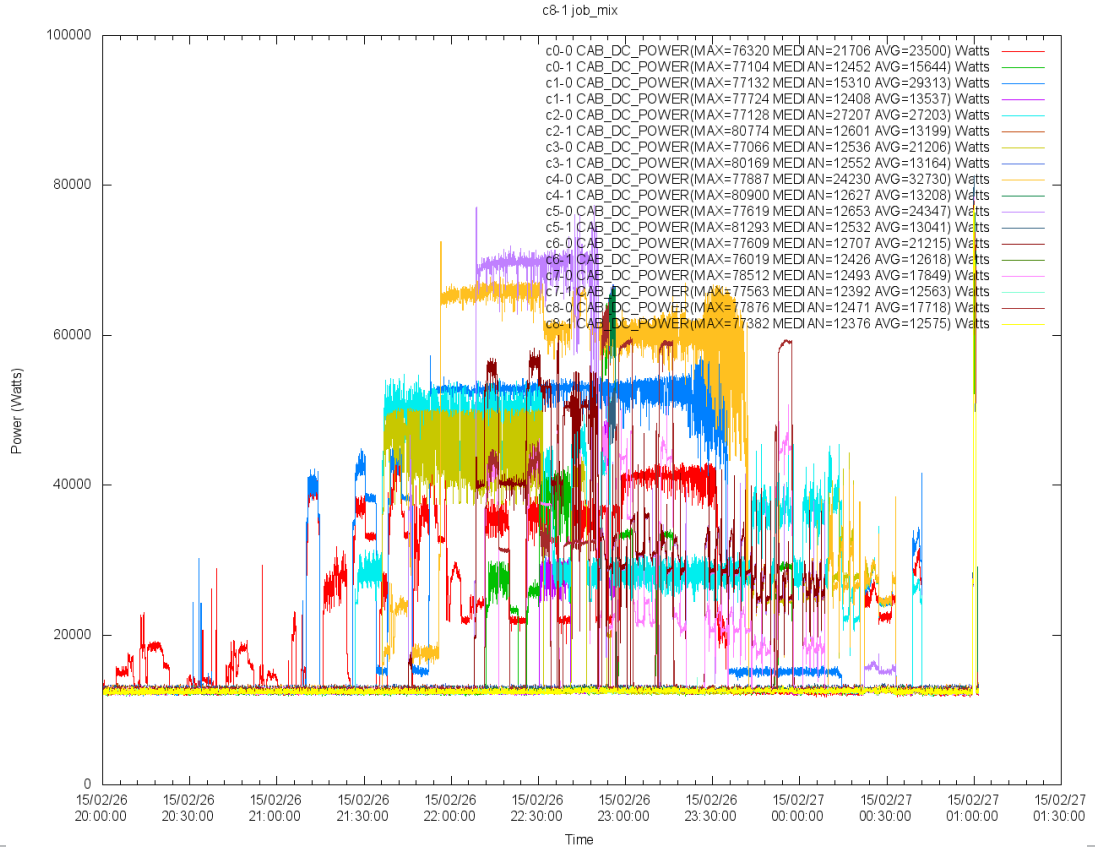
Job Mix: Cabinet Power



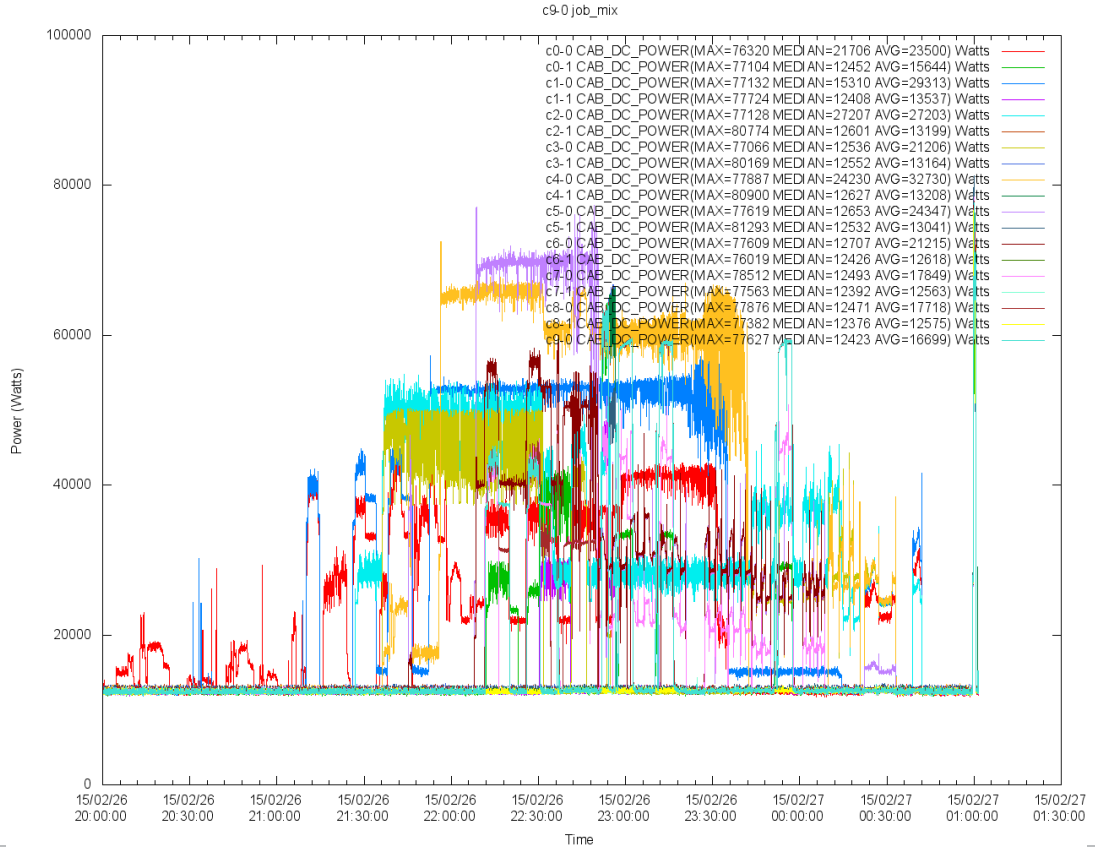
Job Mix: Cabinet Power



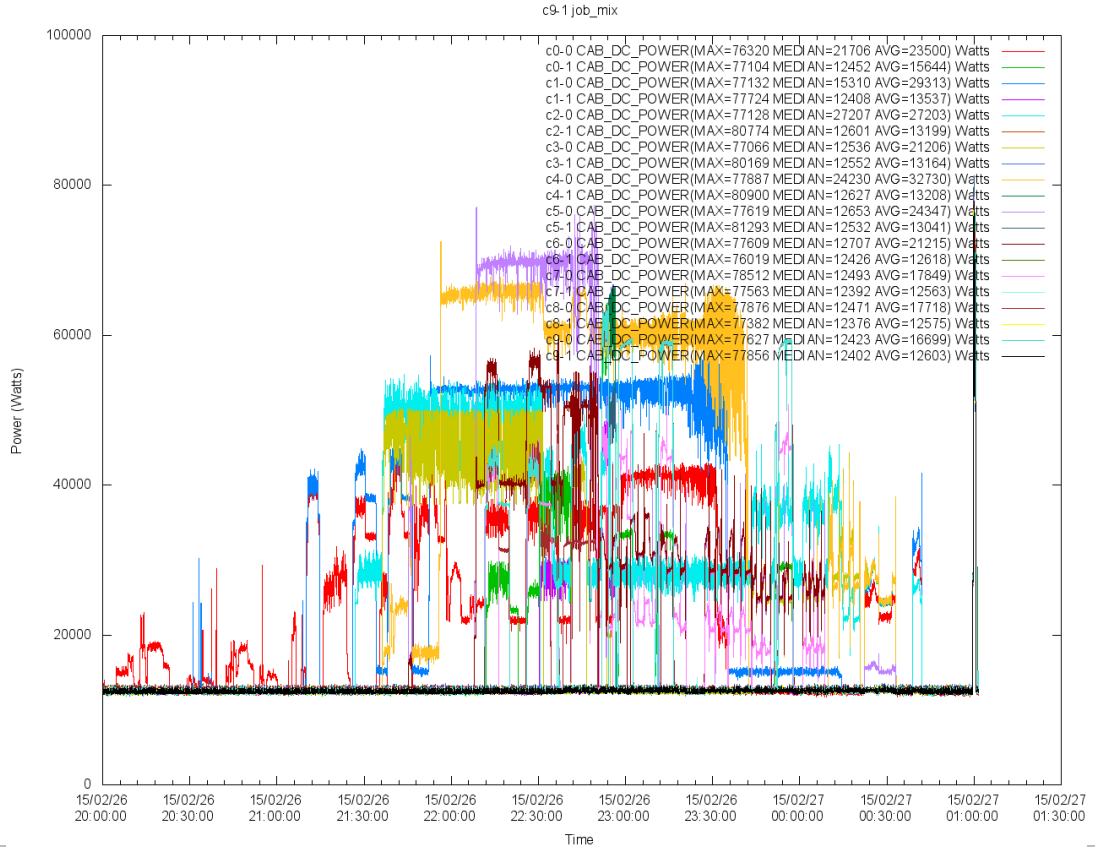
Job Mix: Cabinet Power



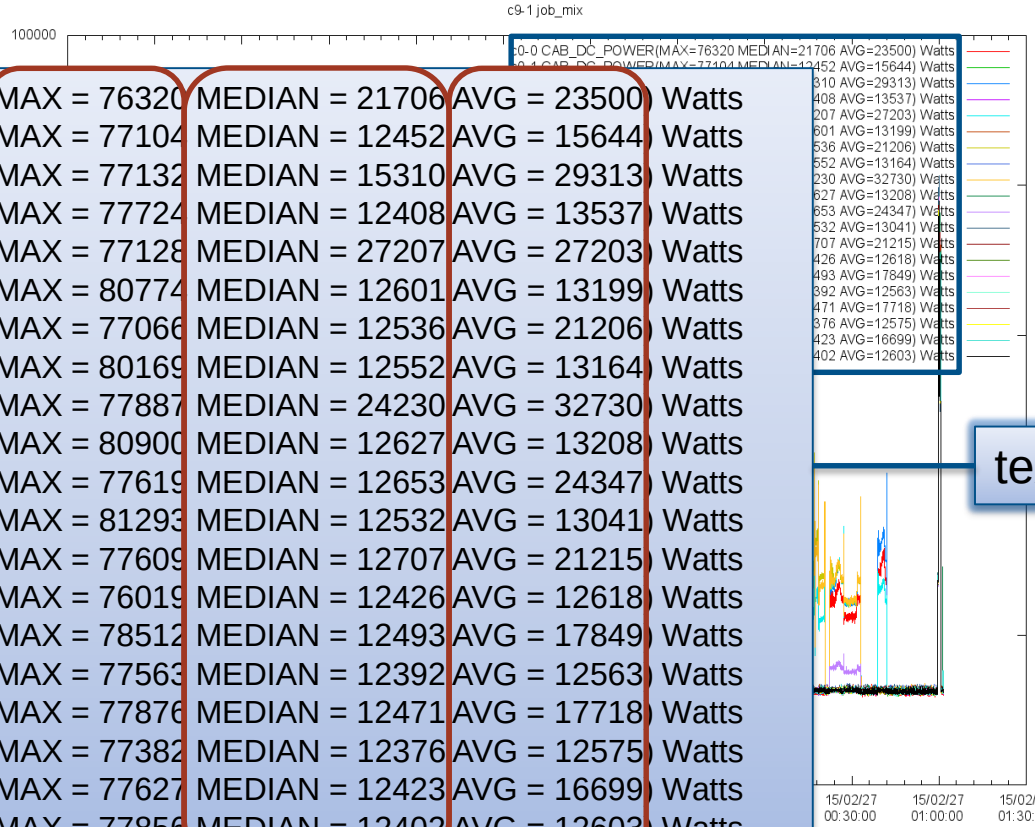
Job Mix: Cabinet Power



Job Mix: Cabinet Power



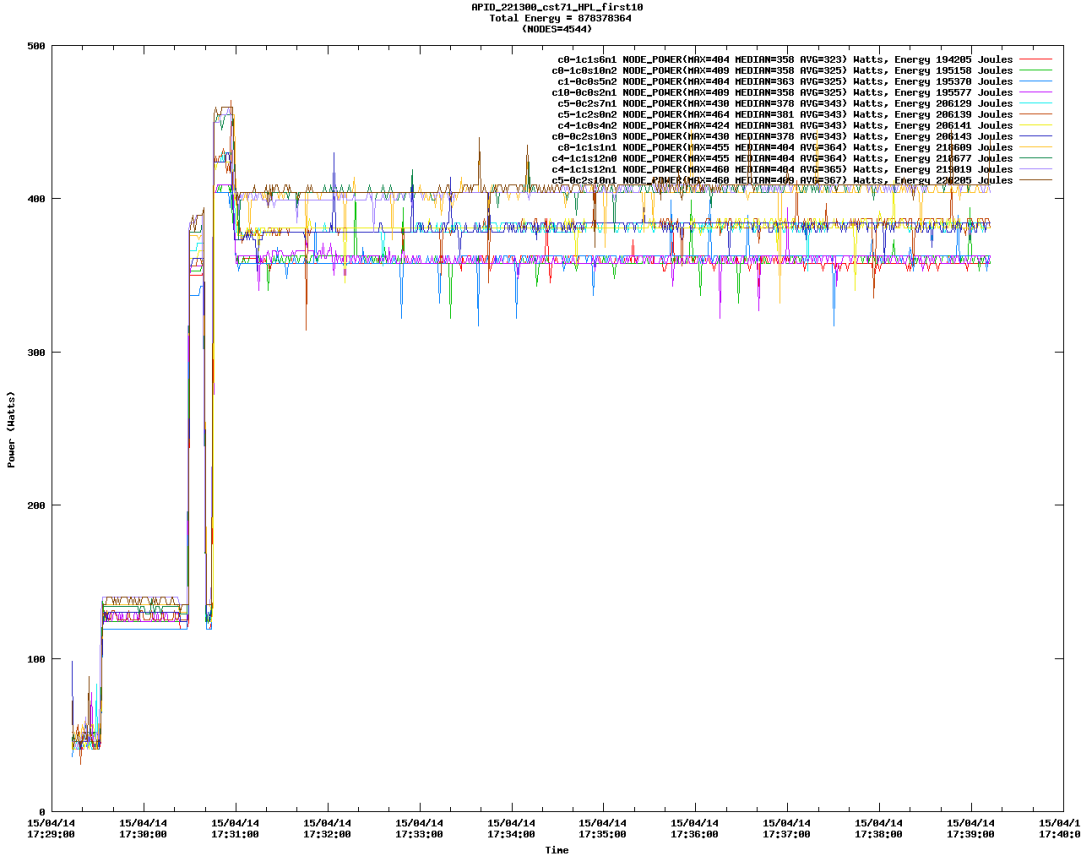
Job Mix: Cabinet Power



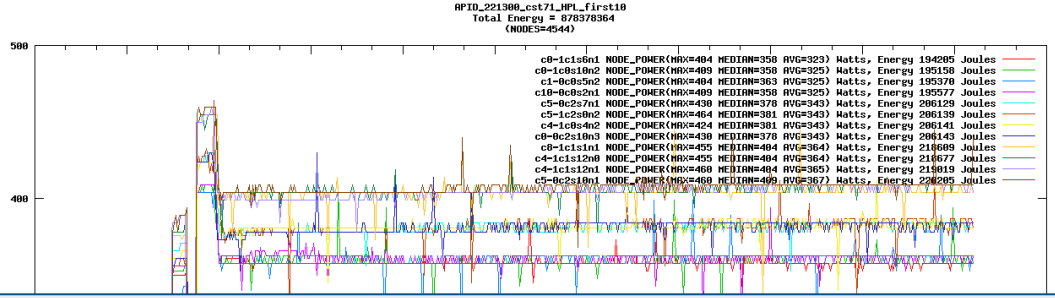
c0-0 CAB_DC_POWER (MAX = 76320 MEDIAN = 21706 AVG = 23500) Watts
c0-1 CAB_DC_POWER (MAX = 77104 MEDIAN = 12452 AVG = 15644) Watts
c1-0 CAB_DC_POWER (MAX = 77132 MEDIAN = 15310 AVG = 29313) Watts
c1-1 CAB_DC_POWER (MAX = 77724 MEDIAN = 12408 AVG = 13537) Watts
c2-0 CAB_DC_POWER (MAX = 77128 MEDIAN = 27207 AVG = 27203) Watts
c2-1 CAB_DC_POWER (MAX = 80774 MEDIAN = 12601 AVG = 13199) Watts
c3-0 CAB_DC_POWER (MAX = 77066 MEDIAN = 12536 AVG = 21206) Watts
c3-1 CAB_DC_POWER (MAX = 80169 MEDIAN = 12552 AVG = 13164) Watts
c4-0 CAB_DC_POWER (MAX = 77887 MEDIAN = 24230 AVG = 32730) Watts
c4-1 CAB_DC_POWER (MAX = 80900 MEDIAN = 12627 AVG = 13208) Watts
c5-0 CAB_DC_POWER (MAX = 77619 MEDIAN = 12653 AVG = 24347) Watts
c5-1 CAB_DC_POWER (MAX = 81293 MEDIAN = 12532 AVG = 13041) Watts
c6-0 CAB_DC_POWER (MAX = 77609 MEDIAN = 12707 AVG = 21215) Watts
c6-1 CAB_DC_POWER (MAX = 76019 MEDIAN = 12426 AVG = 12618) Watts
c7-0 CAB_DC_POWER (MAX = 78512 MEDIAN = 12493 AVG = 17849) Watts
c7-1 CAB_DC_POWER (MAX = 77563 MEDIAN = 12392 AVG = 12563) Watts
c8-0 CAB_DC_POWER (MAX = 77876 MEDIAN = 12471 AVG = 17718) Watts
c8-1 CAB_DC_POWER (MAX = 77382 MEDIAN = 12376 AVG = 12575) Watts
c9-0 CAB_DC_POWER (MAX = 77627 MEDIAN = 12423 AVG = 16699) Watts
c9-1 CAB_DC_POWER (MAX = 77856 MEDIAN = 12402 AVG = 12603) Watts

text_data_job_mix

HPL on 4544 Nodes: First 10 Minutes

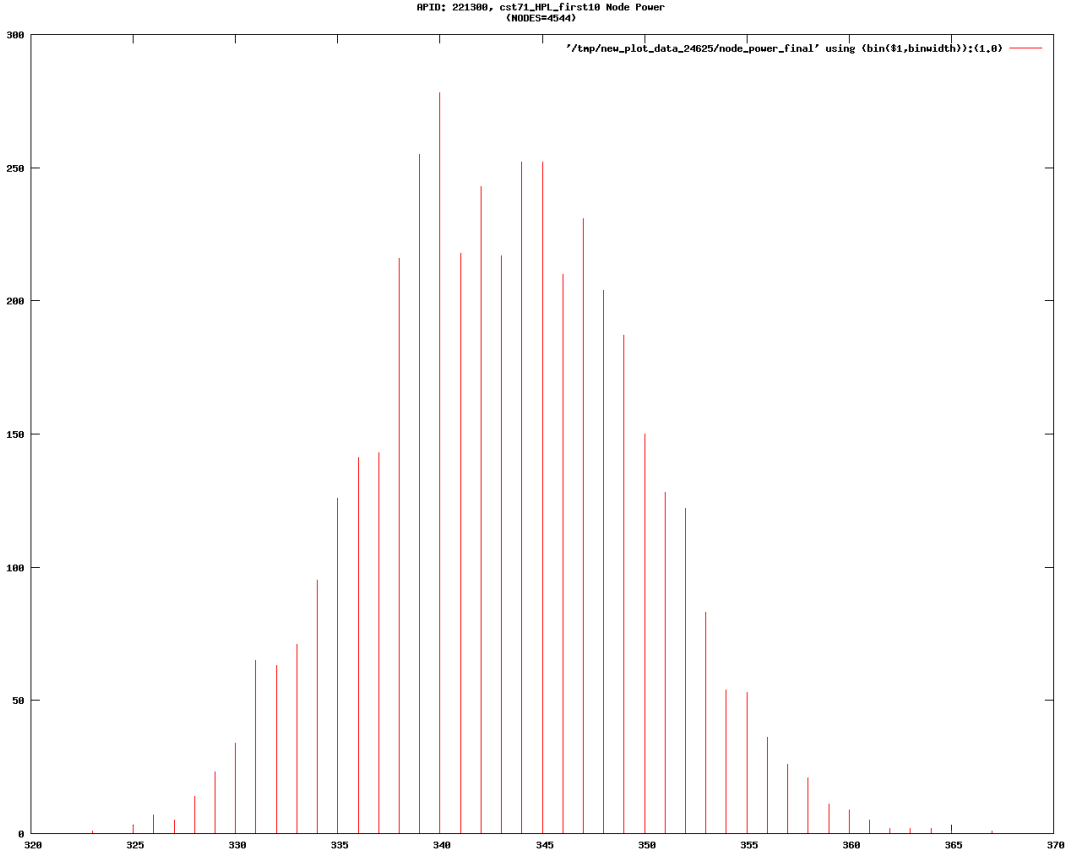


HPL on 4544 Nodes: First 10 Minutes




```
c0-1c1s6n1  NODE_POWER(MAX=404 MEDIAN=358 AVG=323) Watts, Energy 194205 Joules
c0-1c0s10n2 NODE_POWER(MAX=409 MEDIAN=358 AVG=325) Watts, Energy 195158 Joules
c1-0c0s5n2  NODE_POWER(MAX=404 MEDIAN=363 AVG=325) Watts, Energy 195370 Joules
c10-0c0s2n1 NODE_POWER(MAX=409 MEDIAN=358 AVG=325) Watts, Energy 195577 Joules
c5-0c2s7n1  NODE_POWER(MAX=430 MEDIAN=378 AVG=343) Watts, Energy 206129 Joules
c5-1c2s0n2  NODE_POWER(MAX=464 MEDIAN=381 AVG=343) Watts, Energy 206139 Joules
c4-1c0s4n2  NODE_POWER(MAX=424 MEDIAN=381 AVG=343) Watts, Energy 206141 Joules
c0-0c2s10n3 NODE_POWER(MAX=430 MEDIAN=378 AVG=343) Watts, Energy 206143 Joules
c8-1c1s1n1  NODE_POWER(MAX=455 MEDIAN=404 AVG=364) Watts, Energy 218609 Joules
c4-1c1s12n0 NODE_POWER(MAX=455 MEDIAN=404 AVG=364) Watts, Energy 218677 Joules
c4-1c1s12n1 NODE_POWER(MAX=460 MEDIAN=404 AVG=365) Watts, Energy 219019 Joules
c5-0c2s10n1 NODE_POWER(MAX=460 MEDIAN=409 AVG=367) Watts, Energy 220205 Joules
```

HPL on 4544 Nodes: First 10 Minutes



SQL & PMDB

PMDB Database Overview

- **PostgreSQL**
  - PostgreSQL 9.1.12
- **Round-robin data storage…**
  - Configurable: number of partitions, and rows/partition
  - Oldest data (partition) dropped to make room for new…
  - Hook script called when new partitions are created
    - Enables site-level customization

PMDB Database

- **PMDB power & energy**
  - Blade, Node, and Cabinet level power and energy data
  - Job/App ID, start time, end time, and node list data
  - Includes power/energy monitoring for accelerated blades

- **[SEDC system & blade environmental data]**
  - Option
  - Cabinet inlet & outlet water temp and pressure, outlet air temp(s)
  - Blade-level component data (temp, voltage, current)

SEDC → PMDB:
- Option available now, scheduled to become the default in our next major release
- SEDC → flat files will be deprecated, and at some point dropped…

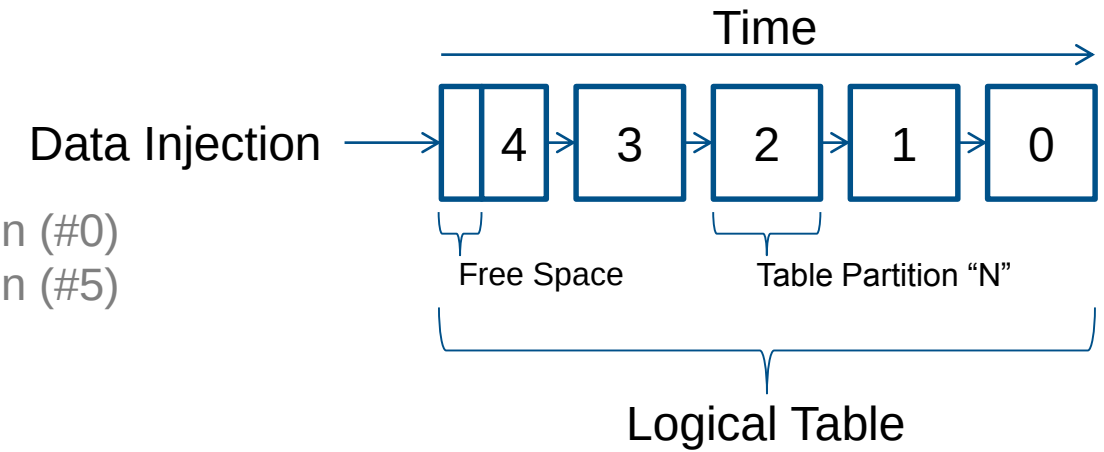
Time Series Data Management

- **Why use SQL database?**
  - Databases store data… (lots of data)
  - Flexible query interface
- **Answer questions**
  - What was max power draw of any node?
  - When did socket temperature reach 50°C?
  - What was average system power last Monday?
- **Difficulties**
  - Data aging
  - Continuous input stream

Data Management

- **Time series data**
  - Continuous stream of input
  - Can't store data indefinitely
  - Age data out using table partitioning

- **Data lifecycle**
  - If no free space
    - Drop oldest partition (#0)
    - Create new partition (#5)
  - Write to free space



Data Management (continued) Table Partitioning

- Define table prototype

```
CREATE TABLE pmdb.bc_data (
    ts TIMESTAMPTZ, source INT,
    id INT, value BIGINT );
```

- Create partitions via inheritance

```
CREATE TABLE pmdb.bc_data_1 ( ) INHERITS (pmdb.bc_data);
```

- Issue queries against “prototype” table

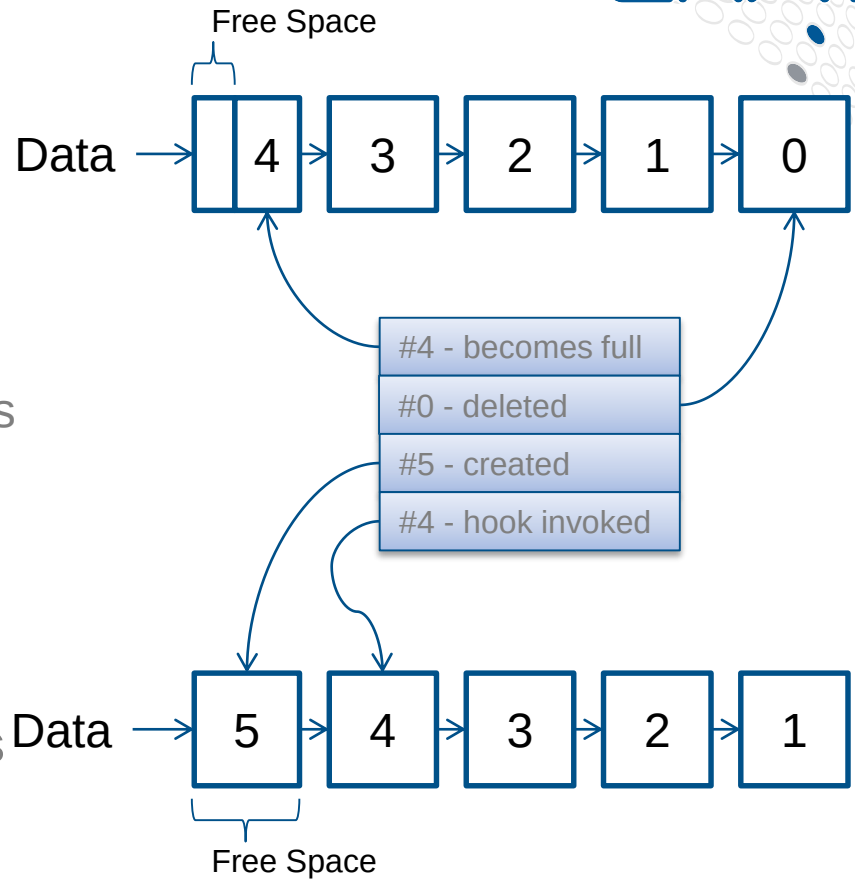
```
SELECT * FROM pmdb.bc_data WHERE …;
```

- “Age out” data by dropping partition & index

```
DROP TABLE pmdb.bc_data_1 CASCADE;
```

Database Hooks (1 of 5)

- **Entry point into PMDB**
  - Provide site customization
  - Can do “anything”
    - Default script (xtpmdbhook.sh) creates indexes & cleans up jobs table
- **“Event” driven**
  - Triggered by partition creation
  - Events for CC & BC data tables

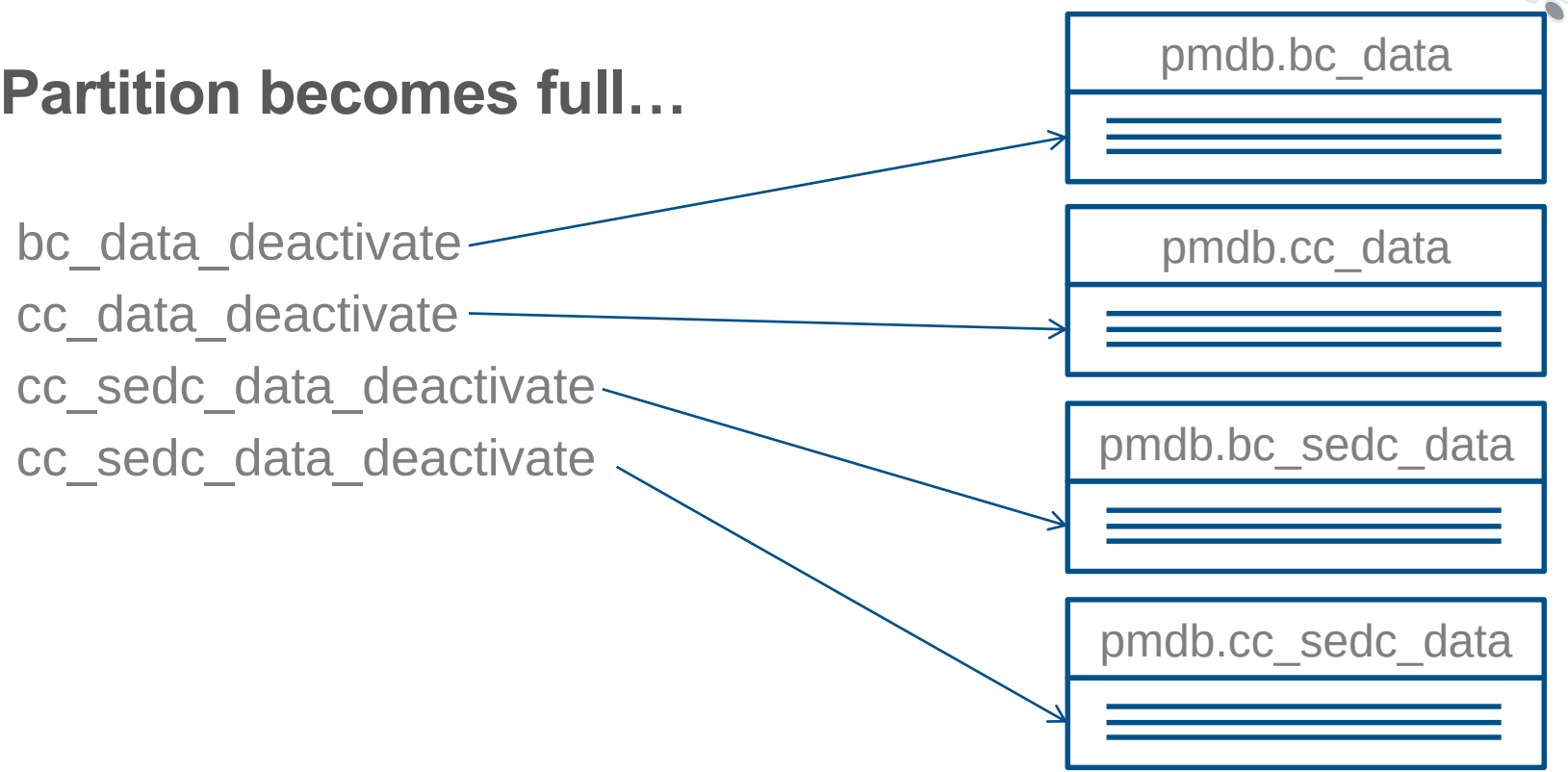


Database Hooks (2 of 5)

- **Calling convention**
  - Command line utility which accepts two arguments
    - Event type
    - Partition name
    - '/path/to/hook-script' <event_name> <partition_name>
  - Output saved to power management log
    `/var/opt/cray/log/power_management-YYYYMMDD`
- **Execution limits**
  - Bounded runtime
    - Default max time → 600 seconds
    - Configurable with 'xtpmdbconfig'
  - No more than 10 concurrent invocations
    - Prevents "fork bombs" if scripts get stuck

Database Hooks (3 of 5) Event types

- **Partition becomes full…**



Database Hooks (4 of 5)

- **Trigger frequency**
  - Depends on data collection configuration
- **Example system**
  - 4 cabinets
    - 48 blades per cabinet
  - pmdb.bc_data ~= 10 data samples per second per blade
    - System wide ~=1920 samples/sec
    - Partition depth → 1,000,000 rows
  - Calculate time to log 1M samples
    - 1,000,000 / 1920 → 521 sec → **8.6 minutes**

Database Hooks (5 of 5) xtpmdbhook.sh

- **Default script on installation**
- **It's more than an example**
  - Cleans up "jobs" tables when blade data ages out
    - Prevents unbounded growth
  - Creates time indexes when partitions fill
    - "pmdb.bc_data"
    - "pmdb.bc_sedc_data"
  - Custom scripts must include this functionality
- **Contains example archiver function**
  - Saves gzipped binary table dump for 1 week

If you replace our default hook you need to deal with the growth of the job_* tables

PMDB: psql pmdb pmdbuser …

```
crayadm@smw:~> psql pmdb pmdbuser
psql (9.1.12)
Type "help" for help.

pmdb=> SELECT * FROM pmdb.cc_data limit 4;
              ts                 |  source   | id |   value
---------------------------------+-----------+----+------------
 2015-04-12 07:46:02.18304-05    | 201326592 |  3 |         252
 2015-04-12 07:46:02.18304-05    | 201326592 |  8 |        5460
 2015-04-12 07:46:02.342361-05   | 205520896 |  0 |       12944
 2015-04-12 07:46:02.342361-05   | 205520896 |  1 | 8121423755
```

PMDB: pmdb.cc_data

```
pmdb=> SELECT ts,source2cname(source) as source,id,value
pmdb-> FROM pmdb.cc_data limit 10;
             ts                    | source | id |   value
-----------------------------------+--------+----+------------
 2015-04-12 07:46:02.18304-05      | c0-0   |  3 |         252
 2015-04-12 07:46:02.18304-05      | c0-0   |  8 |        5460
 2015-0
 2015-0
 2015-04-12 07:46:02.342361-05     | c4-0   |  2 |       51921
 2015-04-12 07:46:02.342361-05     | c4-0   |  3 |         249
 2015-04-12 07:46:02.342361-05     | c4-0   |  8 |        5460
 2015-04-12 07:46:02.388346-05     | c3-0   |  0 |       12582
 2015-04-12 07:46:02.388346-05     | c3-0   |  1 | 6655286020
 2015-04-12 07:46:02.388346-05     | c3-0   |  2 |       51907
```

Cray extension converts the 32bit binary 'source' into a cname string

PMDB: pmdb.bc_data

```
pmdb=> SELECT ts,source2cname(source) as source,id,value
pmdb-> FROM pmdb.bc_data limit 10;
              ts                |   source   | id |  value
--------------------------------+------------+----+----------
 2015-04-14 06:53:03.488398-05 | c0-0c1s3   | 16 |      102
 2015-04-14 06:53:03.488398-05 | c0-0c1s3   | 17 | 43048852
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n0 | 32 |       41
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n0 | 33 | 50018319
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n1 | 40 |       41
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n1 | 41 | 51028737
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n2 | 48 |       41
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n2 | 49 | 51628372
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n3 | 56 |       41
 2015-04-14 06:53:03.488398-05 | c0-0c1s3n3 | 57 | 51447327
```

PMDB: pmdb.job_info & pmdb.job_timing

```
pmdb=> SELECT * FROM pmdb.job_info;
 job_id |  apid   | user_id |     nids
--------+---------+---------+--------------
 339    | 8362460 |   31137 | {764}
 339    | 8362461 |   31137 | {764}
 340    | 8362463 |   31137 | {764,765,766}
 341    | 8362465 |   31137 | {764,765,766}
```

```
pmdb=> SELECT * FROM pmdb.job_timing;
 job_id |  apid   |          start_ts          |          end_ts
--------+---------+----------------------------+----------------------------
 339    | 8362460 | 2015-04-14 09:17:42.850693 | 2015-04-14 09:17:44.919059
 339    | 8362461 | 2015-04-14 09:18:17.636577 | 2015-04-14 09:18:19.524404
 340    | 8362463 | 2015-04-14 09:19:26.458356 | 2015-04-14 09:19:28.689729
 341    | 8362465 | 2015-04-14 09:19:56.234028 | 2015-04-14 09:19:57.266162
```

PMDB: Using pmdb.job_info & pmdb.nodes

```
pmdb=> SELECT * FROM pmdb.job_info;
 job_id |  apid   | user_id |     nids
--------+---------+---------+--------------
 339    | 8362460 |   31137 | {764}
 339    | 8362461 |   31137 | {764}
 340    | 8362463 |   31137 | {764,765,766}
 341    | 8362465 |   31137 | {764,765,766}

pmdb=> SELECT * FROM pmdb.nodes WHERE nid_num in (764,765,766);
   comp_id    | nid_num
--------------+---------
 c3-0c2s15n0 |     764
 c3-0c2s15n1 |     765
 c3-0c2s15n2 |     766
```

PMDB: pmdb.cc_sedc_data

```
pmdb=> SELECT ts,source2cname(source) as source,id,value
pmdb-> FROM pmdb.cc_sedc_data limit 10;
             ts                 | source |  id  | value
--------------------------------+--------+------+-------
 2015-04-11 12:16:48.366747-05 | c2-0   | 1127 |     6
 2015-04-11 12:16:48.366747-05 | c2-0   | 1128 |     0
 2015-04-11 12:16:48.366747-05 | c2-0   | 1129 |   7.5
 2015-04-11 12:16:48.366747-05 | c2-0   | 1130 |   7.5
 2015-04-11 12:16:48.366747-05 | c2-0   | 1131 |   6.5
 2015-04-11 12:16:48.366747-05 | c2-0   | 1132 |   7.7
 2015-04-11 12:16:48.366747-05 | c2-0   | 1133 |   6.5
 2015-04-11 12:16:48.366747-05 | c2-0   | 1134 |   7.3
 2015-04-11 12:16:48.366747-05 | c2-0   | 1135 |   7.7
 2015-04-11 12:16:48.366747-05 | c2-0   | 1136 |   6.2
```

pmdb.cc_sedc_data with pmdb.sedc_scanid_info

```
pmdb=> SELECT ts,source2cname(source) as source,
pmdb-> sensor_name as sensor, value, sensor_units as units
pmdb-> FROM pmdb.cc_sedc_data INNER JOIN pmdb.sedc_scanid_info
pmdb-> ON cc_sedc_data.id=sedc_scanid_info.sensor_id limit 5;
             ts                | source |         sensor          | value | units
-------------------------------+--------+-------------------------+-------+-------
 2015-04-23 10:03:39.766238-05 | c3-0   | CC_T_COMP_CH0_AIR_TEMP0 | 21.49 | degC
 2015-04-23 10:03:39.766238-05 | c3-0   | CC_T_COMP_CH0_AIR_TEMP1 | 21.65 | degC
 2015-04-23 10:03:39.766238-05 | c3-0   | CC_T_COMP_CH0_AIR_TEMP2 | 21.98 | degC
 2015-04-23 10:03:39.766238-05 | c3-0   | CC_T_COMP_CH0_AIR_TEMP3 | 21.32 | degC
 2015-04-23 10:03:39.766238-05 | c3-0   | CC_T_COMP_CH1_AIR_TEMP0 |  23.1 | degC
(5 rows)
```

PMDB: pmdb.bc_sedc_data

```
pmdb=> SELECT ts,source2cname(source) as source,id,value
pmdb-> FROM pmdb.bc_sedc_data limit 10;
             ts                |  source   |  id  |    value
-------------------------------+-----------+------+-------------
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1722 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1723 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1724 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1725 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1726 |     12062.5
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1727 |     12062.5
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1728 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1729 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1730 | 1201.171875
 2015-04-14 05:55:44.8824-05 | c1-0c2s13 | 1731 | 1201.171875
```

PMDB: Find Node Power Sensors Reporting < 10 W

```
crayadm@smw:~> psql pmdb pmdbuser
psql (9.1.12)
Type "help" for help.

pmdb=> SELECT source2cname(source) as cname FROM pmdb.bc_data
pmdb-> WHERE id in (16,32,40,48,56)
pmdb-> and value < 10 and ts > now() - interval '1 minute'
pmdb-> and source2cname(source) in
pmdb-> (select name from sm.expand('rt_node', 's0') where state = 7)
pmdb-> group by cname;
   cname
------------
 c5-0c0s2n0
 c3-0c2s0n1
```

Blade-Level Sensors → pmdb.bc_data

Collected by default on 2-socket Xeon Nodes

ID	Name	ID	Name	ID	Name	ID	Name
16	HSS Power	17	HSS Energy	18	HSS Voltage	19	HSS Current
32	Node 0 Power	33	Node 0 Energy	34	Node 0 Voltage	35	Node 0 Current
40	Node 1 Power	41	Node 1 Energy	42	Node 1 Voltage	43	Node 1 Current
48	Node 2 Power	49	Node 2 Energy	50	Node 2 Voltage	51	Node 2 Current
56	Node 3 Power	57	Node 3 Energy	58	Node 3 Voltage	59	Node 3 Current
64	Node 0 Accelerator Power	65	Node 0 Accelerator Energy	66	Node 0 Accelerator Voltage	67	Node 0 Accelerator Current
72	Node 1 Accelerator Power	73	Node 1 Accelerator Energy	74	Node 1 Accelerator Voltage	75	Node 1 Accelerator Current
80	Node 2 Accelerator Power	81	Node 2 Accelerator Energy	82	Node 2 Accelerator Voltage	83	Node 2 Accelerator Current
88	Node 3 Accelerator Power	89	Node 3 Accelerator Energy	90	Node 3 Accelerator Voltage	91	Node 3 Accelerator Current

Blade-Level Sensors → pmdb.bc_data

Optionally collected

ID	Name	ID	Name	ID	Name	ID	Name
16	HSS Power	17	HSS Energy	18	HSS Voltage	19	HSS Current
32	Node 0 Power	33	Node 0 Energy	34	Node 0 Voltage	35	Node 0 Current
40	Node 1 Power	41	Node 1 Energy	42	Node 1 Voltage	43	Node 1 Current
48	Node 2 Power	49	Node 2 Energy	50	Node 2 Voltage	51	Node 2 Current
56	Node 3 Power	57	Node 3 Energy	58	Node 3 Voltage	59	Node 3 Current
64	Node 0 Accelerator Power	65	Node 0 Accelerator Energy	66	Node 0 Accelerator Voltage	67	Node 0 Accelerator Current
72	Node 1 Accelerator Power	73	Node 1 Accelerator Energy	74	Node 1 Accelerator Voltage	75	Node 1 Accelerator Current
80	Node 2 Accelerator Power	81	Node 2 Accelerator Energy	82	Node 2 Accelerator Voltage	83	Node 2 Accelerator Current
88	Node 3 Accelerator Power	89	Node 3 Accelerator Energy	90	Node 3 Accelerator Voltage	91	Node 3 Accelerator Current

Collected by default on GPU/MIC (accelerated) Nodes

PMDB: pmdb.sensor_info

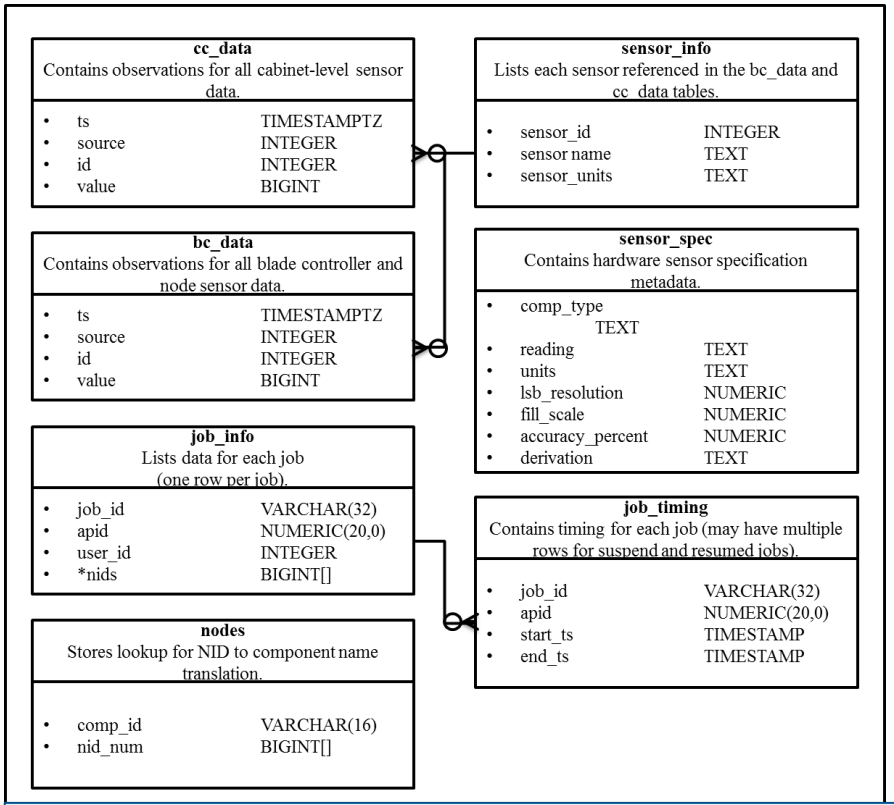
```
pmdb=> SELECT * FROM pmdb.sensor_info ;
 sensor_id |      sensor_name      | sensor_units
-----------+-----------------------+--------------
         0 | Cabinet Power         | W
         1 | Cabinet Energy        | J
         2 | Cabinet Voltage       | mV
         3 | Cabinet Current       | A
         8 | Cabinet Blower Power  | W
        16 | HSS Power             | W
        17 | HSS Energy            | J
        18 | HSS Voltage           | mV
        19 | HSS Current           | mA
        32 | Node 0 Power          | W
        33 | Node 0 Energy         | J
        34 | Node 0 Voltage        | mV
        35 | Node 0 Current        | mA
        40 | Node 1 Power          | W
        41 | Node 1 E              | J
        42 | Node                  | mV
        43 | Node                  | mA
        48 | Node 2                | W
        49 | Node 2                | J
        50 | Node 2 Vo             | mV
        51 | Node 2 Cur            | mA
        56 | Node 3 Powe           | W
        57 | Node 3 Energy         | J
        58 | Node 3 Voltage        | mV
        59 | Node 3 Current        | mA
        64 | Node 0 Accelerato     Power | W
        65 | Node 0 Accelerator Energy   | J
        66 | Node 0 Accelerator Voltage  | mV
        67 | Node 0 Accelerator Current  | mA
        72 | Node 1 Accelerator Power    | W
        73 | Node 1 Accelerator Energy   | J
        74 | Node 1 Accelerator Voltage  | mV
        75 | Node 1 Accelerator Current  | mA
        80 | Node 2 Accelerator Power    | W
        81 | Node 2 Accelerator Energy   | J
        82 | Node 2 Accelerator Voltage  | mV
        83 | Node 2 Accelerator Current  | mA
        88 | Node 3 Accelerator Power    | W
        89 | Node 3 Accelerator Energy   | J
        90 | Node 3 Accelerator Voltage  | mV
        91 | Node 3 Accelerator Current  | mA
```

OOPS!

PMDB: pmdb.sensor_info

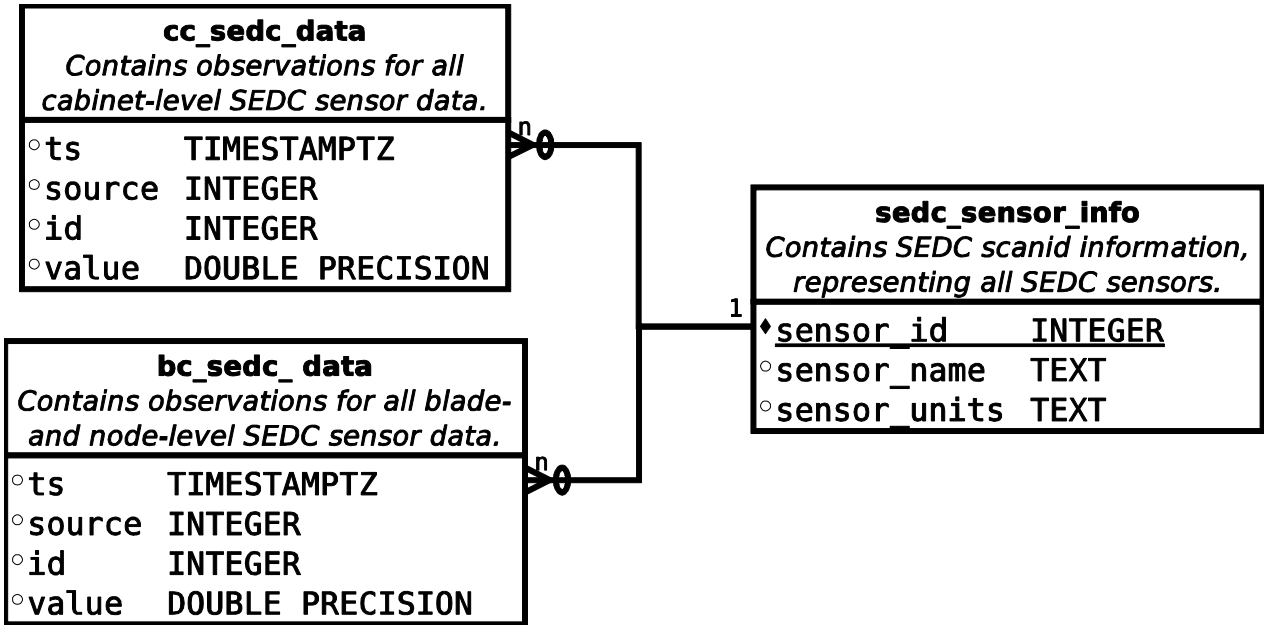
```
pmdb=> SELECT * FROM pmdb.sensor_info ;
 sensor_id |          sensor_name          | sensor_units
-----------+-------------------------------+--------------
         0 | Cabinet Power                 | W
         1 | Cabinet Energy                | J
         2 | Cabinet Voltage               | mV
         3 | Cabinet Current               | A
         8 | Cabinet Blower Power          | W
        16 | HSS Power                     | W
        17 | HSS Energy                    | J
        18 | HSS Voltage                   | mV
        19 | HSS Current                   | mA
        32 | Node 0 Power                  | W
        33 | Node 0 Energy                 | J
```

PMDB Database Schema




```
smw~:/opt/cray/hss/default/etc/xtpmdb.sql
```

PMDB Database Schema, SEDC



PMDB on 4 Cabinet System

```
pmdb=> SELECT pg_database_size('pmdb');
 pg_database_size
------------------
       2955420472
```

```
smw:/var/lib/pgsql # du -h
12K     ./data/pg_multixact/offsets
12K     ./data/pg_multixact/members
28K     ./data/pg_multixact
4.1M    ./data/pg_log
4.0K    ./data/pg_tblspc
4.0K    ./data/base/pgsql_tmp
6.0M    ./data/base/1
2.7G    ./data/base/16385
6.1M    ./data/base/12514
6.0M    ./data/base/12506
2.8G    ./data/base
4.0K    ./data/pg_serial
20K     ./data/pg_subtrans
896K    ./data/global
4.0K
./data/pg_xlog/archive_status
129M    ./data/pg_xlog
4.0K    ./data/pg_twophase
12K     ./data/pg_notify
80K     ./data/pg_stat_tmp
25M     ./data/pg_clog
2.9G    ./data
2.9G    .
```

The Power Database Configuration Tool

```
xtpmdbconfig - The power database configuration tool

Options
-------
    [--help|-h]      Show this text
    [--get|-g]       Get an integer configuration parameter
    [--set|-s]       Set an integer configuration parameter
    [--get-hook|-G]  Get a database hook script
    [--set-hook|-S]  Set a database hook script
    [--show|-i]      Show all configuration parameters
    [--restore|-R]   Restore default configuration parameters
…
```

PMDB on 4 Cabinet System (1 of 3)

```
pmdb=> SELECT min(ts),max(ts),(max(ts)-min(ts)) as time FROM pmdb.bc_data_406;
              min               |              max               |      time
--------------------------------+--------------------------------+-----------------
 2015-03-03 06:58:01.411342-06 | 2015-03-03 07:15:32.104444-06 | 00:17:30.693102
```

```
crayadm@smw:~/>

Showing 9 settings
------------------
bc_max_part_count          = 80
bc_max_part_row_count      = 2000000
bc_sedc_max_part_count     = 80
bc_sedc_max_part_row_count = 2000000
cc_max_part_count          = 100
cc_max_part_row_count      = 100000
cc_sedc_max_part_count     = 50
cc_sedc_max_part_row_count = 100000
hook_max_exec_time         = 600

Showing 4 hooks
------------------
bc_data_deactivate
bc_sedc_data_deactivate
cc_data_deactivate
cc_sedc_data_deactivate
```

00:17:30.693102 * 80 ~= 23 hours

00:54:36.813128 * 80 ~= 72 hours

01:27:42.999999 * 100 ~= 6 days

02:02:06.456031 * 50 ~= 4 days

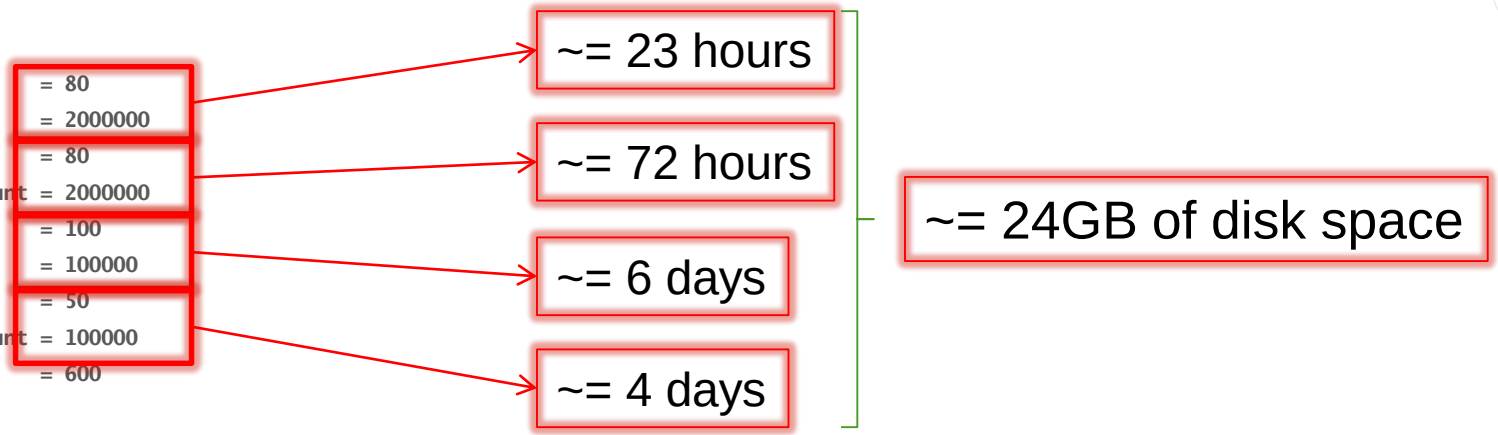
```
pmdb=> SELECT min(ts),max(ts),(max(ts)-min(ts)) as time FROM pmdb.cc_data_63;
              min               |              max               |      time
--------------------------------+--------------------------------+-----------------
 2015-03-02 03:47:10.180329-06 | 2015-03-02 05:14:53.180328-06 | 01:27:42.999999
```

PMDB on 4 Cabinet System (2 of 3)

```
crayadm@smw:~/> xtpmdbconfig --show

Showing 9 settings
------------------
bc_max_part_count          = 80
bc_max_part_row_count      = 2000000
bc_sedc_max_part_count     = 80
bc_sedc_max_part_row_count = 2000000
cc_max_part_count          = 100
cc_max_part_row_count      = 100000
cc_sedc_max_part_count     = 50
cc_sedc_max_part_row_count = 100000
hook_max_exec_time         = 600

Showing 4 hooks
------------------
bc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
bc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
```



PMDB on 4 Cabinet System (3 of 3)

```
smw:/var/lib/pgsql # du -h
12K     ./data/pg_multixact/offsets
12K     ./data/pg_multixact/members
28K     ./data/pg_multixact
3.6M    ./data/pg_log
4.0K    ./data/pg_tblspc
4.0K    ./data/base/pgsql_tmp
6.0M    ./data/base/1
24G     ./data/base/16385
6.1M    ./data/base/12514
6.0M    ./data/base/12506
24G     ./data/base
4.0K    ./data/pg_serial
120K    ./data/pg_subtrans
912K    ./data/global
4.0K    ./data/pg_xlog/archive_status
129M    ./data/pg_xlog
4.0K    ./data/pg_twophase
12K     ./data/pg_notify
124K    ./data/pg_stat_tmp
22M     ./data/pg_clog
24G     ./data
24G     .
```

PMDB on 20 Cabinet System (1 of 3)

```
crayadm@smw:~/> xtpmdbconfig --show

Showing 9 settings
------------------
bc_max_part_count          = 432
bc_max_part_row_count      = 2000000
bc_sedc_max_part_count     = 200
bc_sedc_max_part_row_count = 2000000
cc_max_part_count          = 100
cc_max_part_row_count      = 1000000
cc_sedc_max_part_count     = 50
cc_sedc_max_part_row_count = 200000
hook_max_exec_time         = 600

Showing 4 hooks
------------------
bc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
bc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
```

00:03:33.296673 * 423 ~= 25 hours

00:12:39.431724 * 200 ~= 42 hours

00:18:07.936252 * 100 ~= 30 hours

00:52:34.152681 * 50 ~= 43 hours

PMDB on 20 Cabinet System (2 of 3)

```
crayadm@smw:~/> xtpmdbconfig --show

Showing 9 settings
------------------
bc_max_part_count          = 432
bc_max_part_row_count      = 2000000
bc_sedc_max_part_count     = 200
bc_sedc_max_part_row_count = 2000000
cc_max_part_count          = 100
cc_max_part_row_count      = 1000000
cc_sedc_max_part_count     = 50
cc_sedc_max_part_row_count = 200000
hook_max_exec_time         = 600

Showing 4 hooks
------------------
bc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
bc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_data_deactivate         = /opt/cray/hss/default/bin/xtpmdbhook.sh
cc_sedc_data_deactivate    = /opt/cray/hss/default/bin/xtpmdbhook.sh
```

~= 25 hours

~= 42 hours

~= 30 hours

~= 43 hours

~= 94GB of disk space

PMDB on 20 Cabinet System (3 of 3)

```
smw:/var/lib/pgsql # du -h
4.0K    ./data/pg_xlog/archive_status
129M    ./data/pg_xlog
6.0M    ./data/base/12506
6.1M    ./data/base/12514
6.0M    ./data/base/1
94G     ./data/base/16385
4.0K    ./data/base/pgsql_tmp
94G     ./data/base
2.3M    ./data/pg_log
4.0K    ./data/pg_tblspc
4.0K    ./data/pg_twophase
260K    ./data/pg_stat_tmp
4.2M    ./data/pg_clog
184K    ./data/pg_subtrans
812K    ./data/global
4.0K    ./data/pg_serial
12K     ./data/pg_multixact/members
12K     ./data/pg_multixact/offsets
28K     ./data/pg_multixact
12K     ./data/pg_notify
94G     ./data
94G     .
```

Cray Advanced Platform Monitoring and Control

CAPMC

Come to my paper presentation on Thursday!

CAPMC

- **Available on Cray XC Supercomputer Systems**
  - Available with CLE 5.2.UP02 / SMW 7.2.UP02
- **Primary goal is to enable WLM partners**
  - Access to system power/energy data
  - Access to node-, job-, and app-level power/energy data
  - Access to node-, job-, and app-level power capping controls
    - Accelerator (GPU/MIC) power capping on enabled node types
  - Ability to power off (and on) compute nodes

CAPMC Applets: System-Level Monitoring

- **get_system_power [-s start_time] [-w window]**
  - Returns system-level power data
- **get_system_power_details [-s start_time] [-wwindow_length]**
  - Returns cabinet-level data for all cabinets in the system

Time Format: 'yyyy-mm-dd hh:mm:ss'

CAPMC Applets: Node-Level Monitoring

- **get_node_energy_stats [-s start_time] [-e end_time] **
  [--nids nid_list] [--apid apid] [--jobid job_id]
  - Returns statistics for node-level energy (fixed size response)
- **get_node_energy [-s start_time] [-e end_time] **
  [--nids nid_list] [--apid apid] [--jobid job_id]
  - Returns node-level energy data (one record for each node)
- **get_node_energy_counter -t time [--apid apid] [--jobid job_id] **
  [--nids nid_list]
  - Returns raw accumulated energy counter data (one record for each node)
  - Multiple calls needed, raw counters used for delta calculations

CAPMC Applets: Node Power ON | OFF

- **node_on --nids nid_list**
  - Turn-on nodes and boot Linux making them ready to run jobs
- **node_off --nids nid_list**
  - Shutdown Linux and power off the nodes
- **node_rules**
  - Returns information to the WLM w/respect to node on/off operations
  - Allows System admin to establish constraints
- **node_status [--nids nid_list] [--filter 'opt|opt|opt|...']**
  - Returns current status for requested nodes
  - Allows WLM to poll for status of nodes it is power on/off
  - Filters: show_all, show_off, show_on, show_halt, show_standby, show_ready, show_diag, show_disabled

nid_list: ‘1,3,9-11, 100-300'

CAPMC Applets: Power Capping

- **get_power_cap_capabilities [--nids nid_list]**
  - Returns power capabilities per node-type, for requested nodes
- **get_power_cap [--nids nid_list]**
  - Returns current power cap settings, one record per node
- **set_power_cap --nids nid_list [--node watts] [--accel watts]**
  - Set power cap settings

CAPMC setup on the SMW (1 of 3)

- **As root, copy files:**

```
cp /var/opt/cray/certificate_authority/certificate_authority.crt \
                /etc/opt/cray/capmc/capmc-ca.crt
cp /var/opt/cray/certificate_authority/client/client.key \
                /etc/opt/cray/capmc/capmc-client.key
cp /var/opt/cray/certificate_authority/client/client.crt     \
                /etc/opt/cray/capmc/capmc-client.crt
```

CAPMC setup on the SMW (2 of 3)

- **As root, find the full "hostname" needed for the os_service_url:**
  - cd /var/opt/cray/certificate_authority/hosts
  - snake-smw:~ # openssl x509 -in host.crt -text | grep "CN=" | grep Subject
    Subject: C=XX, ST=XX, O=XX, OU=XX, CN=**snake-smw.us.cray.com**

- **Create capmc.json file (/etc/opt/cray/capmc/capmc.json)**
  - >snake-smw:/etc/opt/cray/capmc # cat capmc.json

```
{
   "os_key":         "/etc/opt/cray/capmc/capmc-client.key",
   "os_cert":        "/etc/opt/cray/capmc/capmc-client.crt",
   "os_cacert":      "/etc/opt/cray/capmc/capmc-ca.crt",
   "os_service_url": "https://snake-smw.us.cray.com:8443"
}
```

CAPMC setup on the SMW (3 of 3)

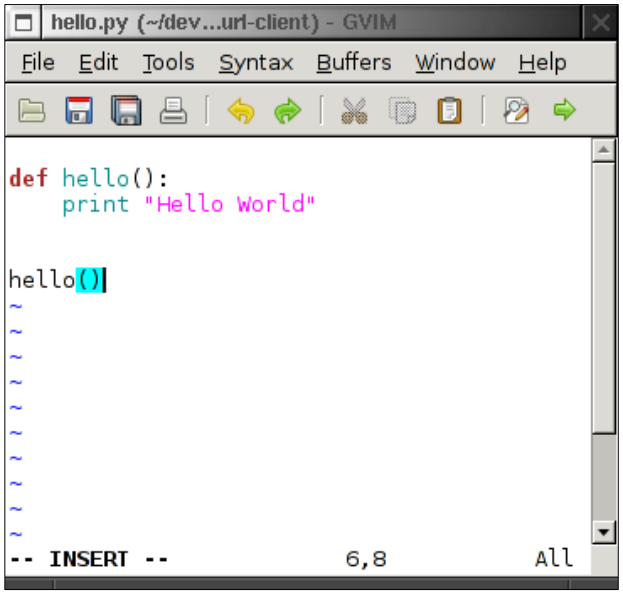
- **# As root, fix file permissions to look like this:**
  - >snake-smw:/etc/opt/cray/capmc # ls -altr

```
total 24
-r--------  1 crayadm crayadm  245 Jul 28  2014 capmc.json
-r--------  1 crayadm crayadm 1123 Jul 28  2014 capmc-ca.crt
-r--------  1 crayadm crayadm  887 Jul 28  2014 capmc-client.key
-r--------  1 crayadm crayadm 3010 Jul 28  2014 capmc-client.crt
drwxr-xr-x  2 root    root    4096 Apr  3 14:00 .
dr-xr-xr-x 15 root    root    4096 Apr  8 22:41 ..
```

Prototype Application

Objective

- **How can we…**
  - Visualize cabinet power
  - Use CAPMC API interface
  - Run on remote white-box
- **Constraints**
  - No direct HTTPS access to SMW
  - Write few lines of code

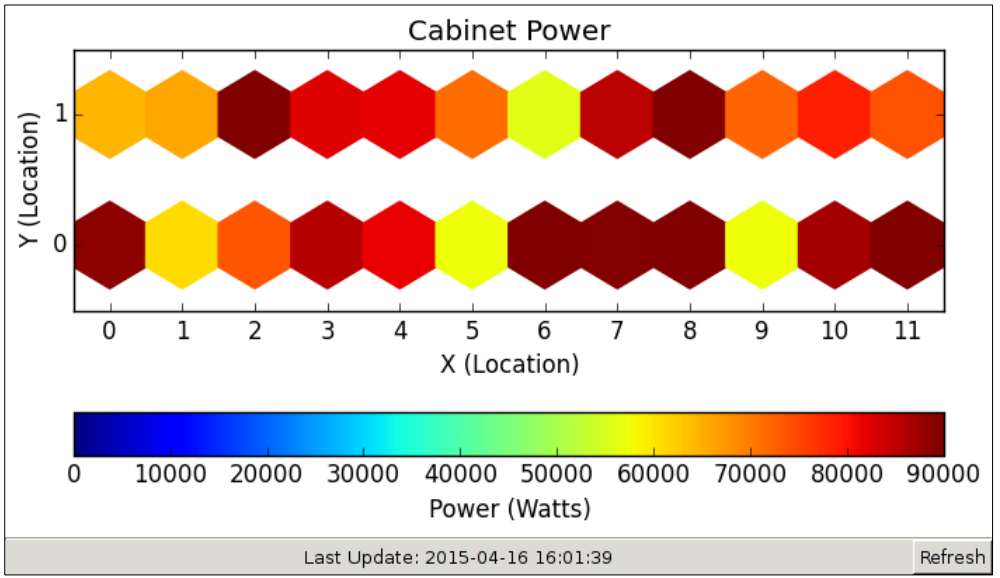


Benefits

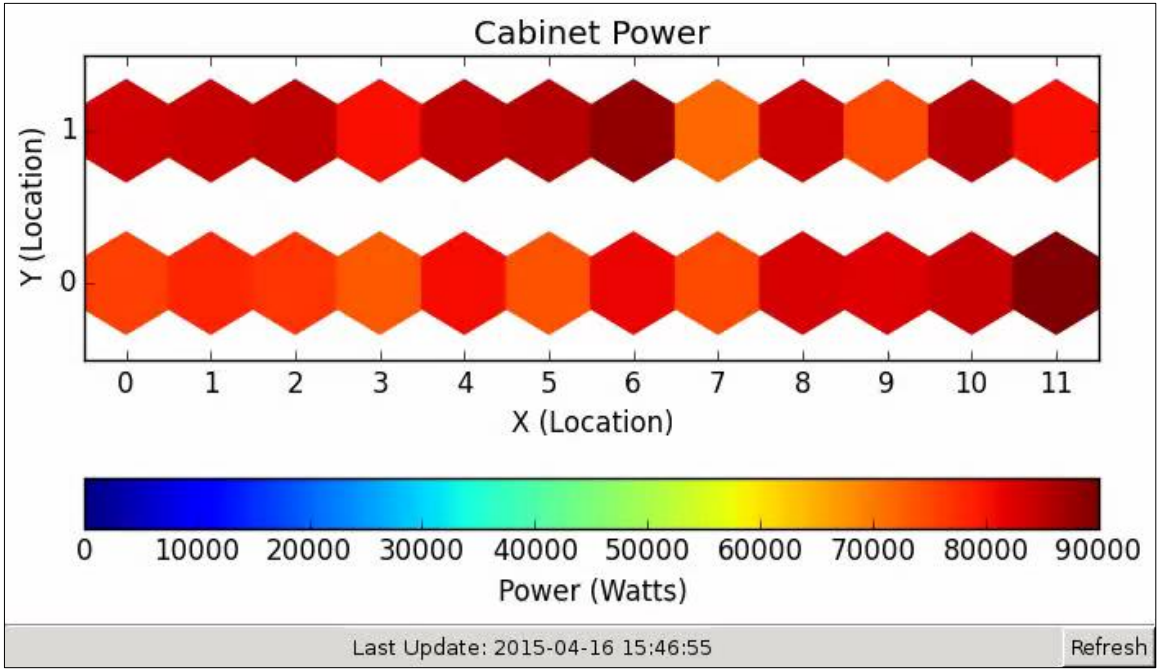
- **Easier data interpretation**
  - “instrument cluster” vs “error log”
- **Visually indicate abnormal behavior**
  - Cabinet emergency power off
    - Power level suddenly drops to zero
  - No applications running
    - Power level near idle state
  - etc…

UI Components

- **Cabinet power data**
  - Each hexagon represents one cabinet
  - X = column, Y = row
  - (4,1) → cabinet c4-1
  - Approximates floor plan
    - Dependent upon cabling
- **Colorized power scale**
- **Indicate "freshness"**
- **Poll every 10 seconds**



Completed Example



(Video playback 40X real time, 24 cabinet system running HPL)

3rd Party Libraries

- **Software Components**
  - Python – http://www.python.org
    - Programming language
  - PyGTK – http://www.pygtk.org
    - Graphics toolkit
  - Matplotlib – http://www.matplotlib.org
    - Data visualization toolkit
  - PyCURL – http://pycurl.sourceforge.net
    - Multi-platform file transfer library

GTK Framework

- **Instantiate UI**
  - Define Python class
  - Implement ‘boilerplate’

```
class BinMapDemo:

    def replot(self):
        # Add acquisition & plotting code here!
        pass

    def timer_tick(self):
        self.replot()
        return True

    def plot_button_cb(self, widget, data=None):
        self.replot()

    def delete_event(self, widget, event, data=None):
        return False

    def destroy(self, widget, data=None):
        gtk.main_quit()
```

```
    def __init__(self):
        self.window = gtk.Window(gtk.WINDOW_TOPLEVEL)
        self.window.connect("delete_event", self.delete_event)
        self.window.connect("destroy", self.destroy)
        self.graph = gtk.Image()
        self.plot_button = gtk.Button("Refresh")
        self.plot_button.connect("clicked", self.plot_button_cb, None)
        self.ulabel = gtk.Label("Last Update: <none>")

        self.vbox = gtk.VBox()
        self.hbox = gtk.HBox()
        self.vbox.add(self.graph)
        self.vbox.add(gtk.HSeparator())
        self.vbox.add(self.hbox)
        self.hbox.add(self.ulabel)
        self.hbox.pack_end(self.plot_button, False, False, 0)
        self.window.add(self.vbox)
        self.window.show_all()
        self.timer_id = gobject.timeout_add(10000, self.timer_tick)
        self.replot()

    def main(self):
        gtk.main()

if __name__ == '__main__':
    demo = BinMapDemo()
    demo.main()
```

Plot Graph

- **Query CAPMC http API**
  - “wish” a query function that returns tuple of new data
  - “get_cab_power()”
- **Matplotlib “hexbin”**
  - Configure range, labels, axes
  - Save plot to temp file
  - Redraw screen with temp file

```
def replot(self):
    (x, y, m) = get_cab_power()
    plt.clf()
    # vmin and vmax set the colorbar range
    plt.hexbin(x, y, gridsize=(max(x), max(y)),
               C=m, vmin=0, vmax=90000)
    plt.axis([-0.5, max(x) + 0.5, -0.5, max(y) + 0.5])
    plt.title("Cabinet Power")
    cb = plt.colorbar(orientation='horizontal')
    cb.set_label("Power (Watts)")
    cb.set_ticks(range(0, 95000, 10000))
    ax = plt.axes()
    ax.set_aspect(1.8)
    ax.set_yticks(range(0, max(y) + 1))
    ax.set_xticks(range(0, max(x) + 1))
    ax.set_xlabel("X (Location)")
    ax.set_ylabel("Y (Location)")
    plt.savefig('cab-power.png',
                bbox_inches='tight', dpi=100)
    self.graph.set_from_file("cab-power.png")
    now = str(datetime.datetime.today()).split('.')[0]
    self.ulabel.set_text("Last Update: %s" % now)
```

Data Query (PyCURL)

- **CAPMC API Query**
  - Post JSON object to URL
  - Munge result
    - JSON array → tuple of lists
  - Still missing http request function

```
def get_cab_power():
    (x, y, m) = ([], [], [])

    data = capmc_post(
            'https://<example-machine>:8443',
            '/capmc/get_system_power_details',
            json.dumps({}),
            '127.0.0.1:8080')

    if isinstance(data, int):
        return (x, y, m)

    data_obj = json.loads(data)
    if "cabinets" in data_obj:
        for cab in data_obj["cabinets"]:
            x.append(int(cab["x"]))
            y.append(int(cab["y"]))
            m.append(int(cab["avg"]))

    return (x, y, m)
```

HTTP Request (PyCURL)

- **HTTP request function**
  - Post text to URL
  - Tell PyCURL about X.509 certificate files
  - Optional SOCKS5 proxy host
  - Return HTTP error code or result body text
- **Only have SSH access?**
  - Tunnel HTTP via SSH proxy

    ssh –D 8081 –N user@remote-system

    socks5h="127.0.0.1:8081"

```
def capmc_post(host, path, post_body, socks5h=None):
    c = pycurl.Curl()
    rx_buffer = StringIO()
    tx_buffer = StringIO(post_body)
    c.setopt(c.URL,      host + path)
    c.setopt(c.CAINFO,   '/path/to/capmc-cacert.pem')
    c.setopt(c.SSLKEY,   '/path/to/capmc-client.key')
    c.setopt(c.SSLCERT, '/path/to/capmc-client.pem')
    c.setopt(c.HTTPHEADER, ['Content-type: application/json'])
    if socks5h != None:
        c.setopt(c.PROXY, 'socks5h://' + socks5h)
    c.setopt(c.POST, 1)
    c.setopt(c.READDATA, tx_buffer)
    c.setopt(c.POSTFIELDSIZE, len(tx_buffer.getvalue()))
    c.setopt(c.WRITEDATA, rx_buffer)
    c.perform()
    sts = c.getinfo(c.RESPONSE_CODE)
    if sts != 200:
        c.close()
        return sts
    body = rx_buffer.getvalue()
    c.close()
    return body
```

Data Query Revisited

- **CAPMC API Query via "capmc" subprocess**
  - Call "capmc" CLI utility
    - No need to write networking code as in previous example
    - CLI is just another program
    - SSH tunneling not supported
  - Munge result
    - JSON array → tuple of lists

```
def get_cab_power():
    (x, y, m) = ([],[],[])

    p = subprocess.Popen(
        ['capmc', 'get_system_power_details'],
        stdout=subprocess.PIPE)
    data = p.stdout.read()
    rc = p.wait()

    if rc != 0:
        return (x, y, m)

    data_obj = json.loads(data)
    if "cabinets" in data_obj:
        for cab in data_obj["cabinets"]:
            x.append(int(cab["x"]))
            y.append(int(cab["y"]))
            m.append(int(cab["avg"]))

    return (x, y, m)
```

Summary

- **Access system telemetry off SMW**
  - Cabinet power query has negligible impact on system operation
- **Utilize software / libraries not available on SMW**
  - Many “useful” Python libraries not shipped on SMW
- **CAPMC built using standards based interfaces**
  - Can use Cray supplied ‘capmc’ CLI client
  - Can use or develop custom 3rd party client

RUR

Resource Utilization Reporting Two Year Update

- **Andrew P. Barry (Cray Inc.) has an RUR paper at CUG this year**
  - Andrew will be presenting on Thursday (Technical Session 17A)
  - Just before my CAPMC talk… ☺.

In the two years since CUG 2013 the Cray RUR feature has gone from PowerPoint to the forth release of software, running on a variety of Cray systems. The most basic features of RUR have proven the most interesting to the widest spread of users: CPU usage, memory usage, and energy usage are enduring concerns for site planning. Functionality added since the first release of RUR has largely focused on providing greater fidelity of measurement, and support for a full range of hardware.

This paper briefly reviews the architecture of the RUR software, describes new functionality added since the initial implementation, and solicits user input on future designs. Also included are a sampling of statistics gathered from Cray datacenter machines contrasted with production machines at Cray customer sites.

Resource Utilization Reporting (RUR)

- **RUR supports a plugin architecture**
  - Many types of data collected using the same infrastructure
- **Several output plugins can be configured**
  - RUR is configured off by default
- **Documented in S-2393:**
  - **“Managing System Software for the Cray Linux Environment”**

RUR Energy Plugin

- **Collects compute node energy usage data**
- **First introduced in CLE 5.0.UP00**
  - One piece of output data: total energy used across all nodes
  - Output data formatted in JSON list format
- **Updated in follow-on CLE releases**
  - Significant numbers of new energy related data points
  - Output data can be formatted in optional JSON dictionary format
  - Maintains backward support for prior JSON list data format

RUR Energy Plugin: Configuration

- **Configure by editing the file:**
  - /etc/opt/cray/rur/rur.conf
  - It's in the shared root, so use xtopview on the boot node

```
...
[plugins]
   gpustat: true
   taskstats: true
   timestamp: true
   energy: true
   memory: false
...
```

```
...
[energy]
stage: /opt/cray/rur/default/bin/energy_stage.py
post:  /opt/cray/rur/default/bin/energy_post.py
arg:   json-dict
...
```

RUR Energy Plugin: Output, json-list

- **Default output format (until CLE 6.0.UP00).**
- **energy_used:**
  - The total energy (joules) used across all nodes
  - On accelerated nodes, this includes energy used by the accelerators

```
2013-08-30T11:19:06.545114-05:00 c0-0c0s2n2 RUR 18657 p2-
20130829t090349 [RUR@34] uid: 12345, apid: 10963, jobid: 0,
cmdname: /scratch/myuser/myapp/bin64/myapp.ex
plugin: energy ['energy_used', 318]
```

RUR Energy Plugin: Output, json-dict (1 of 3)

- **Broader set of metrics, more easily parsed by Python**

- **energy_used:**
  - Total energy (same as 'energy_used‘ in json-list format)
- **nodes:**
  - Number of nodes in job
- **nodes_power_capped:**
  - Number of nodes with nonzero power cap
- **nodes_throttled:**
  - Number of nodes that experienced throttling
    - (e.g., CPU or memory thermal/power throttling)

RUR Energy Plugin: Output, json-dict (2 of 3)

- **Broader set of metrics, more easily parsed by Python**

- **max_power_cap:**
  - Maximum nonzero power cap
- **max_power_cap_count:**
  - Number of nodes with the maximum nonzero power cap
- **min_power_cap:**
  - Minimum nonzero power cap
- **min_power_cap_count:**
  - Number of nodes with the minimum nonzero power cap

RUR Energy Plugin: Output, json-dict (3 of 3)

- Broader set of metrics, more easily parsed by Python

```
2015-01-16T10:23:10.624977-06:00 c0-0c0s1n1 RUR 13513 \
  p1-20150115t070816[RUR@34] uid: 12795, apid: 81870, jobid: 0, \
  cmdname: /bin/hostname, plugin: energy \
{
  "nodes_throttled": 1, "min_accel_power_cap_count": 0,
  "nodes_with_changed_power_cap": 0, "max_power_cap_count": 0,
  "energy_used": 101, "max_power_cap": 0, "nodes_memory_throttled": 1,
  "accel_energy_used": 0, "max_accel_power_cap_count": 0,
  "nodes_accel_power_capped": 0, "min_power_cap": 0,
  "max_accel_power_cap": 0, "min_power_cap_count": 0,
  "min_accel_power_cap": 0, "nodes_power_capped": 0,
  "nodes": 1, "nodes_cpu_throttled": 1
}
```

RUR Energy Plugin: Output Location

- **Lightweight Log Manager (LLM)**
  - Configured as default output plugin.
  - The ‘llm’ plugin writes to file on the SMW:
    - /var/opt/cray/log/partition-current/messages-date
- **Users can**
  - Opt-in for the ‘user’ plugin
  - Redirect plugin output to a specific file or directory
  - Override the default report type, …
  - See **S-2393**: section on “Cray-Supplied Output Plugins”

xtpmaction

SMW command line interface for power monitoring and control

xtpmaction Overview

- **Single point of control for PM operations on the SMW**
  - Enforces policy where needed
- **High level functionality**
  - Power capping setup and management
  - Configuration for power/energy monitoring frequency
  - Configure power threshold setting

xtpmaction -a help

```
crayadm@smw:~> xtpmaction -a help
HELP
    Get help on a specific action
Usage:
    xtpmaction -a help [ACTION]
    ACTIONS:
        activate
        active
        create
        deactivate
        delete
        duplicate
        help
        list
        power
        power_overbudget_action
        properties
        propinfo
        pscan
        rename
        reset
        show
        supported
        sysinfo
        system_power_threshold
        update
        validate
        xtinfo
```

Power Capping with xtpmaction (1 of 2)

- **Create a profile**
  - `xtpmaction --action create --percent 100 \`
    `--profile cap_100`
- **Customize your profile**
  - `xtpmaction -a power --profile cap_100 -i`
- **Activate the profile**
  - `xtpmaction -a activate --profile cap_100`

Power Capping with xtpmaction (2 of 2)

- **List all profiles**
  - `xtpmaction -a list`
- **Show the active profile**
  - `xtpmaction -a show`
- **Deactivate the profile**
  - `xtpmaction -a deactivate [--profile cap_100]`

xtpmaction -a help create

CREATE

Create a power profile based on a percentage of the available host range (max - min) of compute nodes. Default profile name will be '__THRESH%%.pX‘ where the percentage will be specified instead of '%%'. If no percentage is specified, 100 percent will be used. Power caps will be applied to the node control for compute nodes only. Accelerators (if present) will not be power capped by this function, unless power availability constraints are such that a limit on accelerator power use is required.
The profile name to create can be specified with a '--profile' option

ARGS:

-q flag may be used to reduce verbosity
--force flag may be used to overwrite target profile if it exists

Usage:

```
xtpmaction -a create [-q][--force][--partition PARTITION] \
      [--percent PERCENTAGE][--profile PROFILE]
```

Examples:

```
xtpmaction --action create --percent 80 --profile THRESH_PROFILE
xtpmaction --action create --percent 80 --partition p0
```

xtpmaction --action create --percent 100 \ --profile cap_100

```
crayadm@cst72:~/stevem> xtpmaction --action create --percent 100 --profile cap_100

Profile: /opt/cray/hss/default/pm/profiles/p0/cap_100.p0

Descriptor                                  Limits          #Nodes   %node   %host   %accel
#(ComputeANC_HSW_270W_24c_64GB_2133_NoAccel)
compute|01:000d:306f:010e:0018:0040:0855:0000 node=415        566      100     100     0
#(ComputeANC_HSW_270W_24c_64GB_2133_NoAccel)
service|01:000d:306f:010e:0018:0040:0855:0000 node=0          2        0       0       0   (no power cap)
#(Service_SNB_115W_8c_32GB_14900_NoAccel)
service|01:000a:206d:0073:0008:0020:3a34:0000 node=0          4        0       0       0   (no power cap)
crayadm@cst72:~/stevem>
```

Demo Live?

xtpmaction -a help power

POWER

Show a total system power estimate. If no profile specified, the current active profile is used, or if no profile is active, a profile will be auto-generated with a node limit of 100 percent. The partition must be specified if multiple partitions exist, unless the partition is part of the specified power profile name. If the percentage argument is prefaced with a +/- sign, then this is understood as a percentage increase/decrease of the current percentage of node limit range. The '--percent_increase' and '--percent_decrease' arguments specify a percentage by which to increase or decrease the current node limit percentage. For example, a power profile with an 80 percent node limit would become a 90 percent node limit if the '--percent_increase 10' argument was provided. The '--powered' argument will restrict the power estimate to nodes that are currently powered on. By default, all nodes, powered or unpowered are included in the power estimate. The '-noff/--num_off' argument specifies number of compute nodes to assume are powered off. The '-i/--interactive' argument specifies that the power estimate should run in interactive mode. When run in interactive mode the user is shown a menu of choices for altering the power estimate, re-displaying the power estimate, or generating a power profile from the estimate.

Arguments:

```
    [-p/--partition PARTITION][--percent_increase PERCENTAGE] [-f/--profile PROFILE]   [--powered]
    [-P/--percent PERCENTAGE] [--percent_decrease PERCENTAGE] [-noff/--num_off NUMBER] [-i/--interactive]
```

Examples:

```
   xtpmaction -a power --profile   fullspeed.p2
   xtpmaction -a power --partition p2 --profile fullspeed
   xtpmaction -a power --partition p2
   xtpmaction -a power --profile   fullspeed -P 80
   xtpmaction -a power --profile   fullspeed -P -20 -noff 5
   xtpmaction -a power --profile   fullspeed -i --powered
   xtpmaction -a power --profile   fullspeed -i -P 80
   xtpmaction -a power --profile   fullspeed --percent_increase 10
   xtpmaction -a power --profile   fullspeed --percent_decrease 10
```

xtpmaction --action power --profile cap_100 -i

```
crayadm@cst72:~/stevem> xtpmaction --action power --profile cap_100 -i

Estimated power use for profile: cap_100.p0

  Sub total:        234890 Num:   566 Pwr:     415 100% Max: 415 (compute|ComputeANC_HSW_270W_24c_64GB_2133_NoAccel)
  Sub total:           830 Num:     2 Pwr:     415 100% Max: 415 (service|ComputeANC_HSW_270W_24c_64GB_2133_NoAccel)
  Sub total:           740 Num:     4 Pwr:     185 100% Max: 185 (service|Service_SNB_115W_8c_32GB_14900_NoAccel)
Profile total:      236460
  Sub total:         14600 Num:   146 Pwr:     100 Static blade power
  Sub total:         18000 Num:     3 Pwr:    6000 Static cabinet power
  Sub total:             0 Num:     1 Pwr:       0 Static system power
Static total:        32600
Combined total:     269060      Current system peak power use:   232352

Choose an option:
         1) percentage
         2) percentage increase
         3) percentage decrease
         4) percentage increase and descriptor to apply increase to
         5) percentage decrease and descriptor to apply decrease to
         6) watts and descriptor to apply setting to
         7) number of nodes assumed powered off
         8) number assumed off and descriptor to apply power off assumption to
         9) use powered nodes only
        10) use powered/unpowered nodes
        11) show power estimate
        12) create power profile
Choice: ('q' to quit) [1-12]: 6
('c' to cancel) [watts,descriptor]: 350,compute|ComputeANC_HSW_270W_24c_64GB_2133_NoAccel
watts,descriptor: 350,compute|ComputeANC_HSW_270W_24c_64GB_2133_NoAccel
```

Demo Live?

xtpmaction --action power --profile cap_100 -i

```
Choose an option:
         1) percentage
         2) percentage increase
         3) percentage decrease
         4) percentage increase and descriptor to apply increase to
         5) percentage decrease and descriptor to apply decrease to
         6) watts and descriptor to apply setting to
         7) number of nodes assumed powered off
         8) number assumed off and descriptor to apply power off assumption to
         9) use powered nodes only
        10) use powered/unpowered nodes
        11) show power estimate
        12) create power profile
Choice: ('q' to quit) [1-12]: 12
Choice: ('c' to cancel) [cap_100.p0]: cap_350w
Created profile: /opt/cray/hss/default/pm/profiles/p0/cap_350w.p0
```

```
Choose an option:
         1) percentage
         2) percentage increase
         3) percentage decrease
         4) percentage increase and descriptor to apply increase to
         5) percentage decrease and descriptor to apply decrease to
         6) watts and descriptor to apply setting to
         7) number of nodes assumed powered off
         8) number assumed off and descriptor to apply power off assumption to
         9) use powered nodes only
        10) use powered/unpowered nodes
        11) show power estimate
        12) create power profile
Choice: ('q' to quit) [1-12]: q
crayadm@cst72:~/stevem>
```

Demo Live?

xtpmaction -a power --profile cap_100

566 compute nodes, each with a cap set at 415 watts

146 Blades, w/static power of 100 watts/blade

3 cabinets, 4000 watts/cab

```
crayadm@cst72:/opt/cray/hss/default/pm> xtpmaction --action power --profile cap_100

Estimated power use for profile: cap_100.p0

  Sub total:        234890 Num:   566 Pwr:     415 100% Max: 415 (compute|ComputeANC_HSW_270W_24c_[illegible]133_NoAccel)
  Sub total:           830 Num:     2 Pwr:     415 100% Max: 415 (service|ComputeANC_HSW_270W_24c_64GB_2133_NoAccel)
  Sub total:           740 Num:     4 Pwr:     185 100% Max: 185 (service|Service_SNB_115W_8c_32GB_14900_NoAccel)
Profile total:      236460
  Sub total:         14600 Num:   146 Pwr:     100 Static blade power
  Sub total:         12000 Num:     3 Pwr:    4000 Static cabinet power
  Sub total:             0 Num:     1 Pwr:       0 Static system power
Static total:        26600
Combined total:     263060       Current system peak power use:   224668
```

Demo Live?

Worst-case power estimate

xtpmaction -a help validate

VALIDATE

Validate a profile, partition, properties file or all of the above.

If the 'all' argument is specified, the profile and partition options are ignored

Usage:

```
xtpmaction -a validate [--profile PROFILE] \
    [--partition PARTITION][all]
```

Examples:

```
xtpmaction -a validate --profile PROF.p0
xtpmaction -a validate --partition p0
xtpmaction -a validate properties
xtpmaction -a validate all
```

xtpmaction -a help activate

ACTIVATE

Activate a specified profile.

Usage:

```
xtpmaction -a activate [--partition PARTITION] --profile PROFILE
```

Examples:

```
xtpmaction -a activate --profile PROF.p0
xtpmaction -a activate --profile PROF    --partition p0
xtpmaction -a activate --profile PROF.p2 --partition p2
```

xtpmaction -a help deactivate

DEACTIVATE

Deactivate a specified profile for a specified optional partition.
Deactivate the current active profile if no profile is specified
If no partition suffix specified on profile, the command
attempts to add one (i.e. PROF -> PROF.p0)

Usage:

```
xtpmaction -a deactivate [--partition PARTITION][--profile PROFILE]
```

Examples:

```
xtpmaction -a deactivate
xtpmaction -a deactivate -p p0
xtpmaction -a deactivate --profile PROF.p0
xtpmaction -a deactivate --profile PROF --partition p0
```

xtpmaction -a help delete

DELETE

Delete specified profile. If no partition suffix on profile name, the command will attempt to add one. If the active profile is deleted, it will be deactivated

ARGS:

-q flag may be used to reduce verbosity

Usage:

```
xtpmaction -a delete [-q] [--partition PARTITION] --profile PROFILE
```

Examples:

```
xtpmaction -a delete --profile PROF.p0
xtpmaction -a delete --partition p2 --profile PROF
xtpmaction -a delete -q --profile PROF.p0
```

xtpmaction -a help list

LIST

List available profiles.

If no partition specified, list all profiles available in all partitions

Usage:

```
xtpmaction -a list [--partition PARTITION]
```

Examples:

```
xtpmaction -a list
xtpmaction -a list --partition p2
```

xtpmaction -a help pscan

PSCAN

Set power management scanning frequencies Allow setting both a system scan rate and a high frequency rate.
The scan frequency will be cached for subsequent invocations. Valid frequency values are:

#	Integer system scan period in milliseconds ([1000-10000])
#	Integer hf scan period in milliseconds ([200-10000])
on	Period gets system default value
off	Turn off scanning

If no scan frequency value is specified, the previously stored scan value will be used if it exists.
The optional 'show' keyword will display the currently cached scan settings

ARGS:

-n MODULE_LIST (a list of modules on which to apply scan settings)
-q flag may be used to reduce verbosity
-c flag may be used to reset any cached settings to their default values.

Usage:

```
xtpmaction -a pscan [-q] [-c] [--partition PARTITION] [--system-scan SysScanFreq]\
      [--hf-scan HighFreqScanFrequency][-n modulelist][-N modulelist_file] [show]
```

Examples:

```
xtpmaction -a pscan -q --system-scan on --hf-scan off
xtpmaction -a pscan -q --hf-scan 333 -n 'c0-0c0s0,c0-0c0s1'
xtpmaction -a pscan -q --hf-scan 333 -n 'c0-0c0s0,c0-0c0s1' --system-scan off
xtpmaction -a pscan -q --partition p0 --hf-scan on -n 'c0-0c0s0,c0-0c0s1'
xtpmaction -a pscan --partition p0 --hf-scan 333 -N /tmp/MODULELIST_FILE
xtpmaction -a pscan --partition p0 show
```

Using xtpmaction -a pscan

Show current settings

```
crayadm@cst72:~/stevem> xtpmaction -a pscan show
Partition:                  p0
Accel Sensors:              0x030303030303030300030000
Non-Accel Sensors:          0x303030300030000
Queue Time:                 5
System Scan Period:         1000ms
High Freq Scan Period:      200ms
High Frequency Module List:     []
crayadm@cst72:~/stevem>
```

Using xtpmaction -a pscan

Enable high frequency scanning in one blade

```
crayadm@cst72:~/stevem> xtpmaction -a pscan --hf-scan on -n c0-0c1s4
Checking cached system scan settings...
CMD: cat /tmp/tmpHgCasF|xtpscan --start \
        --sensor=0x303030300030000 --queue-time=5 --period=1000 -N -
CMD: cat /tmp/tmpy
        --sensor=0
```

```
crayadm@cst72:~/stevem> xtpmaction -a pscan show
Partition:                  p0
Accel Sensors:              0x03030303030303030030000
Non-Accel Sensors:          0x303030300030000
Queue Time:                 5
System Scan Period:         1000ms
High Freq Scan Period:      200ms
High Frequency Module List:     ['c0-0c1s4']
```

xtpmaction -a help power_overbudget_action

POWER OVERBUDGET ACTION:

Get or set the node power overbudget action on all blades.
'get' will display the current overbudget action. 'set' will update the
Supported overbudget actions are:

- log (log the event) (this is the default act
- nmi (halts the node and drops t
- power_off (powers off the no

Usage:

```
X                                       tion ['get'|'set' ACTION]

                         _overbudget_action
                   power_overbudget_action get
        tion -a power_overbudget_action set log
 xtpmaction -a power_overbudget_action set nmi
 xtpmaction -a power_overbudget_action set power_off
```

CAUTION: Be aware that applying either of the non-default actions above will bring down nodes and cause applications to fail (If an over-budget condition is detected). We strongly recommend that before changing the default action you review the log messages carefully and consult with Cray Service Personnel for alternative solutions.

Backup Slides

Steven J. Martin ..... (stevem@cray.com)
David Rush ..... (rushd@cray.com)
Matthew Kappel ..... (mkappel@cray.com)

Additional Resources

“Monitoring and managing power consumption on the Cray XC30 system”

- **Cray S-0043-72**
- **http://docs.cray.com/books/S-0043-7203/S-0043-7203.pdf**

“Managing system software for the Cray Linux Environment”

- **Cray S-2393-52xx**
- **http://docs.cray.com/books/S-2393-5203/S-2393-5203.pdf**

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, URIKA, and YARCDATA. The following are trademarks of Cray Inc.: ACE, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.