Disclaimer

• What follows is for existing products or experimental work
• It is not a commitment to future products
Recent History:
Blue Gene/Q, SuperMUC, POWER 775

- All machines used direct-water-cooling of above-ambient water for a majority of the compute rack power

- All machines operated with a constant temperature and flow inlet water supply provided by a coolant distribution unit (CDU)
  - Filtered to not plug quick-connects and fine pitch fins
  - Treated with biocides and corrosion inhibitors

- POWER 775 had integrated in-rack CDUs, SuperMUC at LRZ had custom 1 MW CDUs, Blue Gene/Q used both commercial rack-sized CDUs and facility level CDUs
  - For Blue Gene/Q, created fast-acting shutoff valve to protect against leaks. See next slide.

- All machines had a (correlated) choice of inlet flow rate and inlet water temperature
  - See slide 4 for conceptual allowed operating envelope.
  - Some clients would seasonally change inlet water temperature (and flow) to stay within envelope

- POWER 775 and Blue Gene/Q could measure water temperature and flow.
  - Periodically stored in a database along with device temperatures
  - Accuracy depended on measurement method
Example: Flow Rate vs. T1 (inlet water temp)
Example: Flow Rate vs. T2 (Outlet Temp)

T1: inlet water temperature to Rack
T2: return water temperature to facility

Assume using $T_{1_{\text{worst}}}$ to Rack

Dew point $+1^\circ\text{C}$

Water Temperature ($^\circ\text{C}$)

Water Flow Rate (GPM)
Facility Considerations

• Previous curves were for maximum power
  – Facility is free to change flow rate or temperature to stay within envelope
  – Additionally, inlet water temperatures can always be reduced as dew-point falls.

• Compute racks will vary in power, depending on application
  – Maximum envelop is still valid

• If all nodes were doing the same thing, then:
  – Envelopes change as a function of rack power but that can be pre-computed
  – If it were desired to operate with as hot an exit water temp as possible (to maximize “free cooling”) then for direct cooling loop could just regulate to exit water temperature

• But all nodes may not be doing the same thing, which makes fine-grained regulation problematic.
  – In general, IBM goes not recommend fine-grained regulation of temperature or flow.
Energy Aware Scheduler (EAS)

• IBM Platform Computing Job Scheduler (LSF) can “learn” job attributes and manage CPU, GPU power.
  – Finds total energy, runtime, and maximum power as a function of CPU frequency
  – Controls CPU and GPU power states through configurable policy (minimal time, or minimal total energy, or ...)
  – If no information is available (first time job is run) then prudent assumptions is it will draw maximum power
  – This job information is used for energy aware scheduling (manage to power caps, etc)
  – Currently in prototype stage using X86 CPUs.
Energy Aware Scheduling Policies

**Idle Nodes:**

Policy Driven Power Saving
- Suspend the node to the S3 state (saves ~60W)
- Suspend/Hibernate via xCat
- Idle for a configurable period of time.
- Policy windows (i.e. 22:00 – 07:00)
- Site customizable to use other suspension methods

- Power Saving Aware Scheduling
  - Schedule jobs to use idle nodes first (Power saved nodes as last resort)
  - Aware of job request and wake up nodes precisely on demand
  - Safe period before running job on resumed nodes

- Manual management
  - Suspend, resume, history

**Active Nodes:**

- Ability to set the node/core frequency for a given job/application/user.

- Intelligent prediction of performance, power consumption and runtime of applications at different frequencies

- Energy Saving Policies
  - Save energy with a degradation <= X% (lower freq)
  - Minimize the time to Solution (raise freq)

- Collection of the power usage for an application (AC and DC)

- Scheduling thresholds based on other environmental factors – such as node temperature
Power Capping Policy - Example Configuration

• Power is cheaper overnight, so allow more power to be used at night, and less during the day when it is more expensive

• Add definition of the Power Capping policy to the lsb.threshold file, for example:

  Begin PowerCap
  POWERCAP TIMEWINDOW
  500  (17:00-08:00)
  300  ()
  End PowerCap

• NOTES:
  – POWERCAP values must be in kilowatts (kW).
  – TIMEWINDOW values must follow a 24 hour clock.
  – POWERCAP with an empty TIMEWINDOW - () - is treated as a default.
Power Capping Policy - Submitting Jobs

• The Administrator enables the power capping policy

• Submit a job, which uses an energy tag, for example:

  # bsub -x -a "eas(mytag, minimize_time)" a.out
  Job <307> is submitted to queue <admin>.

• Check the job's information:

  # bjobs -l
  Job <307>, User <test>, Project <default>, Status <RUN>, Queue <admin>, Combined
  CPU Frequency <2.30 GHz(auto)>, Energy policy tag <test.mytag>, Command <a.out>
  Tue Feb 3 21:21:32: Submitted from host <idb3c21>, CWD </home>, Exclusive Execution, Re-
  runnable;
  Tue Feb 3 21:21:32: Started on <idb3c20>, Execution Home </home/test>, Execution CWD </home>;

  ... EXTERNAL MESSAGES:
  MSG_ID FROM POST_TIME MESSAGE ATTACHMENT
  0 - - - -
  1 test Feb 3 21:21:32 POWERCAP[power=192.065491] N

• The job will only be dispatched if it will not violate the power cap policy.