SAND2014-2591C



A Use Case Approach to Deriving Power API Requirements

Sue Kelly and Jim Laros
Sandia National Laboratories
Ryan Elmore, Steve Hammond, and Kris Munch
National Renewable Energy Laboratory

MoabCon 2014

April 2, 2014

Sandia National Laboratories

Exceptional service in the national interest

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Power API – *A Use Case Approach*

Sandia National Laboratories

- Reviewed by Labs, Universities and Commercial partners

Lawrence Livermore National Laboratory

NREL
NATIONAL RENEWABLE ENERGY LABORATORY

Los Alamos
NATIONAL LABORATORY
EST. 1943

NERSC

PENGUIN COMPUTING
EXPERTS IN HIGH PERFORMANCE COMPUTING SOLUTIONS

Georgia Tech

UNM

AMD

CRAY
THE SUPERCOMPUTER COMPANY

OAK RIDGE
National Laboratory

hp

EEHPC
ENERGY EFFICIENT HIGH PERFORMANCE COMPUTING

Truth in Advertising…

- Why I am here
  1. Garner support for a common HPC power API
  2. Persuade you that our effort on the right track
  3. Give (what I believe and hope is) an interesting technical talk to a community interested in Resource Managers and Schedulers

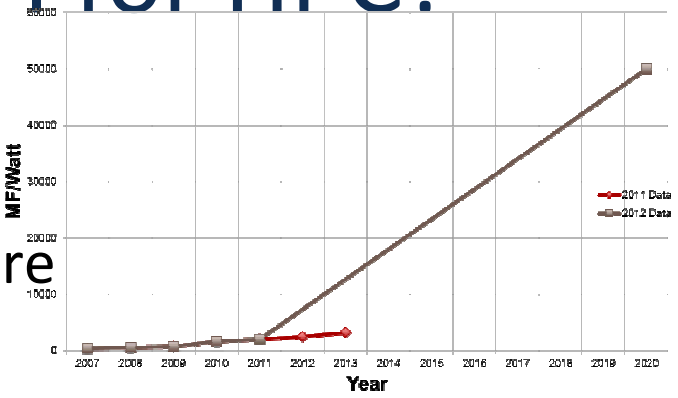
Outline

- Motivation for a Power API
- Our Power API Use Case Model
- Status of our Power API Specification (one slide)
- Sampling of HPC power research by Use Case Actor/System pairs

Why Develop a Power API for HPC?

Sandia National Laboratories

- Anticipated HPC computational needs within reasonable power constraints require significant advances in hardware power efficiency



- To achieve greatest efficiency, software at many levels will need to coordinate and optimize the hardware power features
  - Commodity pressures will drive useful innovations
  - Our efforts are distinguished by HPC requirements at scale
  - We found no existing portable API specification
- Goal is to create a *generic, comprehensive*, power API for general adoption within the HPC community
- Our intent is to
  - help ASC field machines with reasonable power budgets
  - contribute to the national effort on extreme scale scientific computing

Example Scenarios needing an API

Sandia National Laboratories

- A job is entering a checkpoint phase. Application requests a reduced processor frequency during the long I/O period.
- Developer is trying to understand frequency sensitivity of an algorithm and starts a tool that analyzes performance and power consumption while the job is running.
- Data Center has a maximum of capacity of nn MW. One HPC system is down for extended maintenance. Other systems can have a higher maximum power cap.
- Power company charges more for electricity during the day than it does at night. Schedule jobs by allocating both node and power resources accordingly.

Power API – *A Use Case Approach*

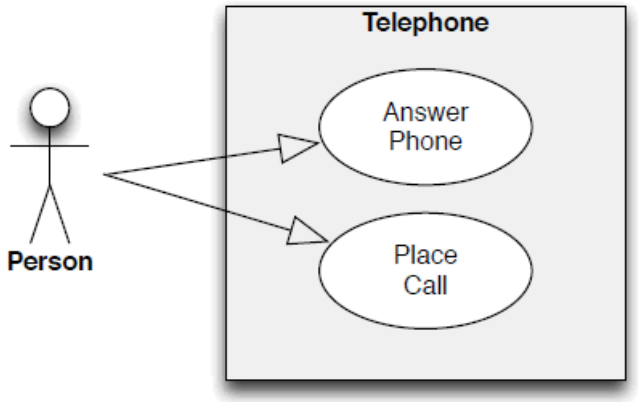
Sandia National Laboratories

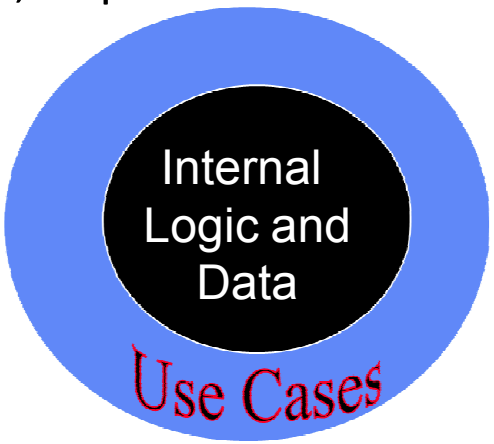
- Prior to specifying an API from scratch, we elected to create formal use cases[1] to model the ways power measurement and control capabilities will be used in HPC systems.
- Use case approach is used to define **SCOPE, INTERFACES,** and **DATA REQUIREMENTS.**
- Generally more intuitive than a laundry list of specifications.

[1]Ivar Jacobson. *Object Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.

Use Case Concepts taken from UML specification ISO/IEC 19501:2005

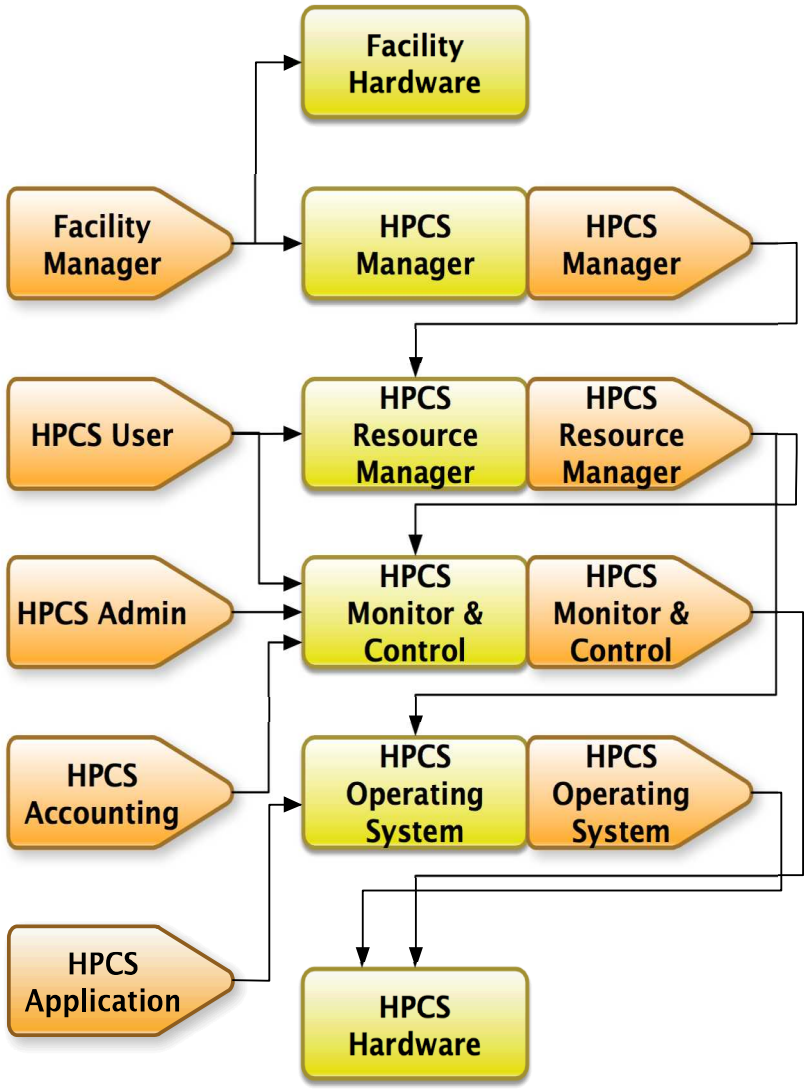
- Use Case – A specific way of ***using*** the system by performing some part of the functionality.
- Actor – A representation of what interacts with the system. May be a person, another system, or something else (e.g. cron or async event).
- Use cases are represented by ovals. Typical naming convention is a verb followed by object. Subject is implied by the initiating actor.
- An actor is represented by a stick figure.
- There is no notion of where data resides or its layout. It's like a floating platter that one gets data off of, or puts data on.



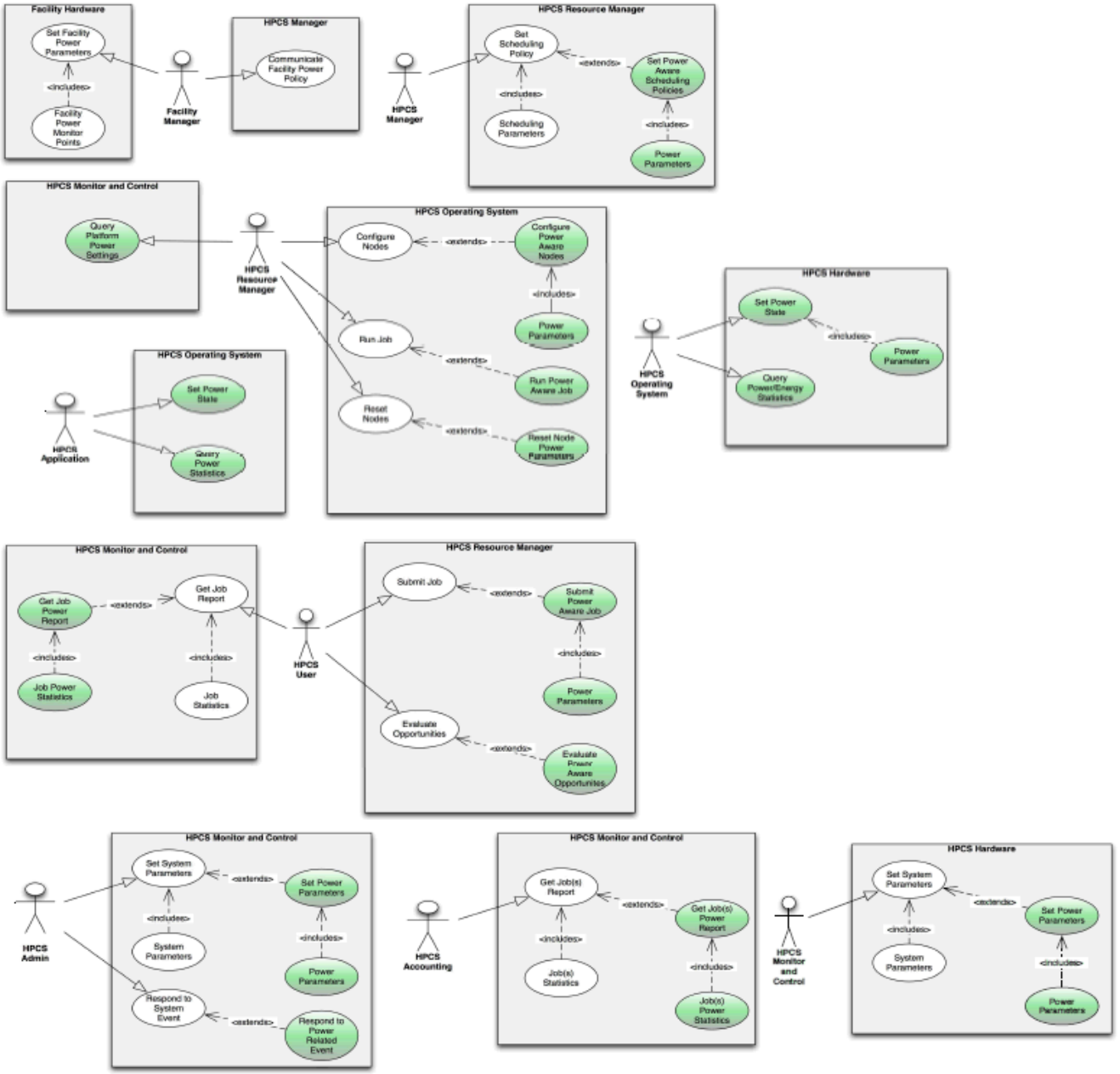


Power API Actors and Systems

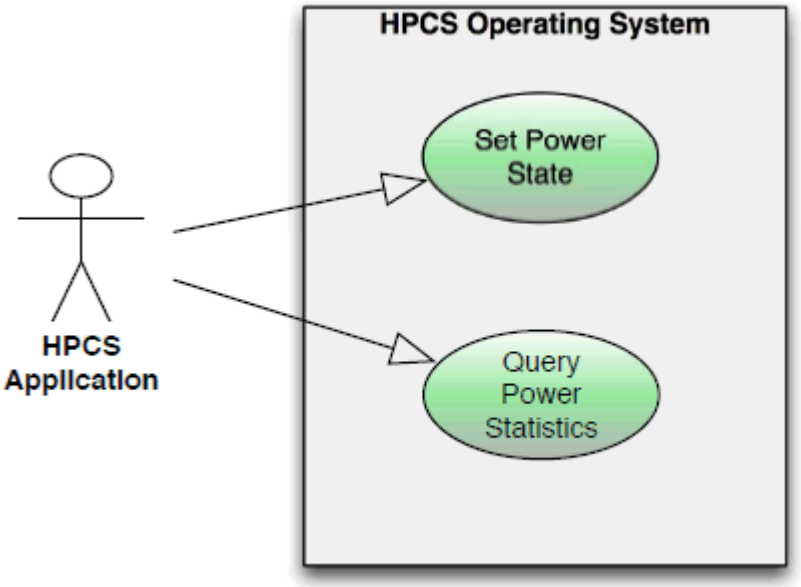
- A high level view of the entire scope to be covered by the API
- A system can also be an actor
- Resource Manager includes scheduler
- A Runtime System is not called out. Implemented in these Systems
  - Resource manager(s)
  - Application (esp. Libraries)
  - Operating Systems



Our Use Case Model



Actor: HPCS Application
System: HPCS Operating System



Actor	HPCS Application
System	HPCS Operating System
Use Case	Set Power State
Description	While an application is running, one, some, or all of the processes may choose to actively manage its power or energy consumption on its node. For example, it may request to lower power state on the CPU while writing a checkpoint to disk or going into an I/O intensive phase.
Trigger	The application reaches a point in the code where it knows that the power state is either particularly important or not important.
Flow of Events	1. As a precursor to setting a new power state, the application requests information on available power states and the current power state of its components (see Query Power Statistics use case 3.10.2). 2. The operating system returns requested power/energy state and statistics. 3. If the current power state is not optimum, the application decides how much to raise or lower the power/energy state of one or more components. 4. The application requests that the power state be changed on one or more components. 5. The operating system returns success along with state information.
Alternative Paths	5a. The operating system returns failure along with state information. 5b. The Application can decide to resubmit request immediately or wait until a later point in execution.
Frequency	Could be measured in seconds, more likely in minutes
Input Data	In step 4, the application requests a power state change.
Output Data	In response to step 2, the available power states are provided in some generic, system portable fashion.
Pre Condition	Application is executing at power/energy configuration A
Post Condition	Application is executing at power/energy configuration B
Power API	yes

Power API specification is started

Sandia National Laboratories

PowerAPI_SAND.pdf - Adobe Acrobat Pro

File Edit View Document Comments Forms Tools Advanced Window Help

3.3 Application, Operating System Interface

```
1 typedef enum {
2     PWR_SLEEP_NO
3     PWR_SLEEP_SHALLOW
4     PWR_SLEEP_MEDIUM
5     PWR_SLEEP_DEEP
6     PWR_SLEEP_DEEPEST
7 } PWR_SleepState
8
9 int PWR_SetDeepestSleepState(PWR_Obj, PWR_SleepState)
```

PWR_SetDeepestSleepState allows the application to request that the OS restricts the deepest sleep state that the hardware can enter. This is not required to be honoured by the OS or HW, but serves as a hint to the OS that low latency sleep to active transitions are desired. Calling this on an object that does not provide sleep states will return an error PWR_ERR_NOT_PROVIDED.

```
1 int PWR_WakeUpLatency(PWR_Obj, PWR_SleepState)
```

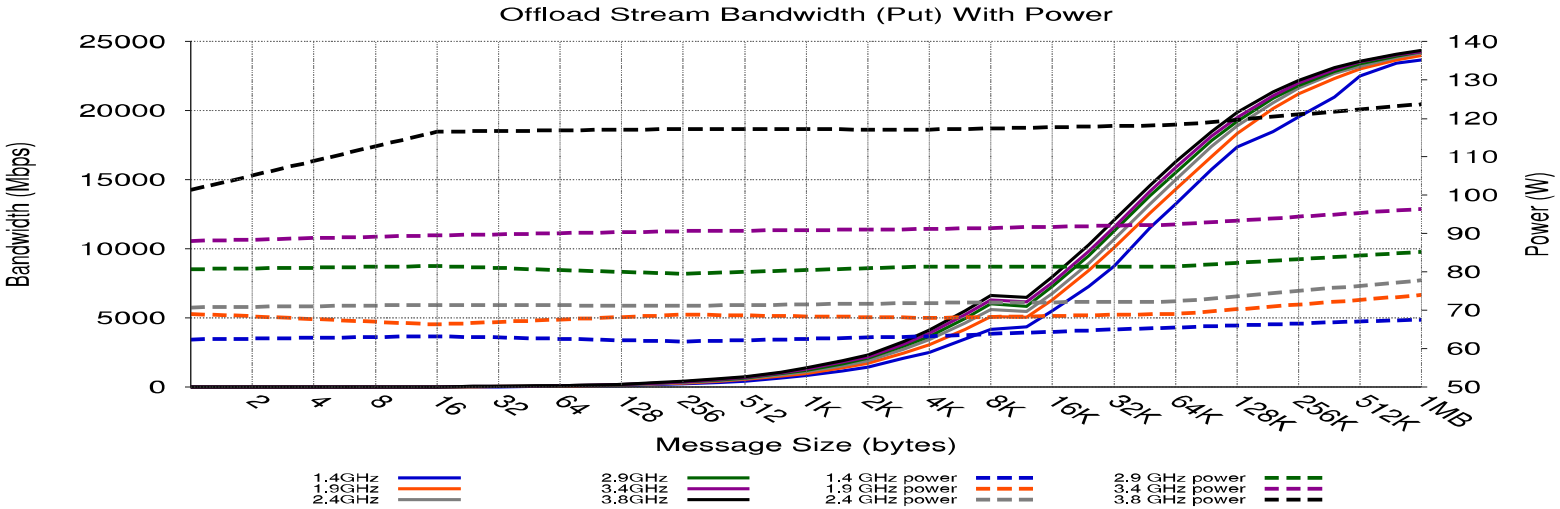
PWR_WakeUpLatency provides the application with the opportunity to determine the latency of waking up from a given sleep state. This information can be used to provide a better value to PWR_SetDeepestSleepState based on the application's sensitivity to latency.

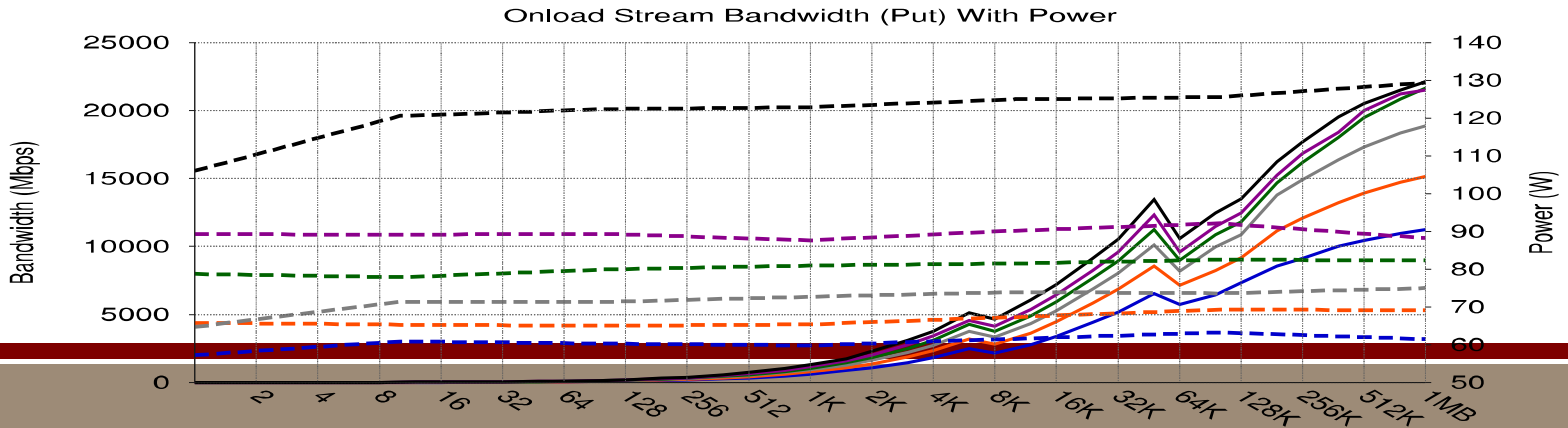
```
1 PWR_GetPEState(); get and set call for some abstracted Power and/or Energy
      State, needs to be fleshed out
2 PWR_SetPEState()
```

```
1 PWR_GetActiveCoreCount();
2 /* get number of cores active (not powered down and available for use by
     the application) */
3 PWR_GetCoreState();
4 /* get state of a core (or group of cores?) */
```

Actor: HPC Application
System: HPCS Operating System

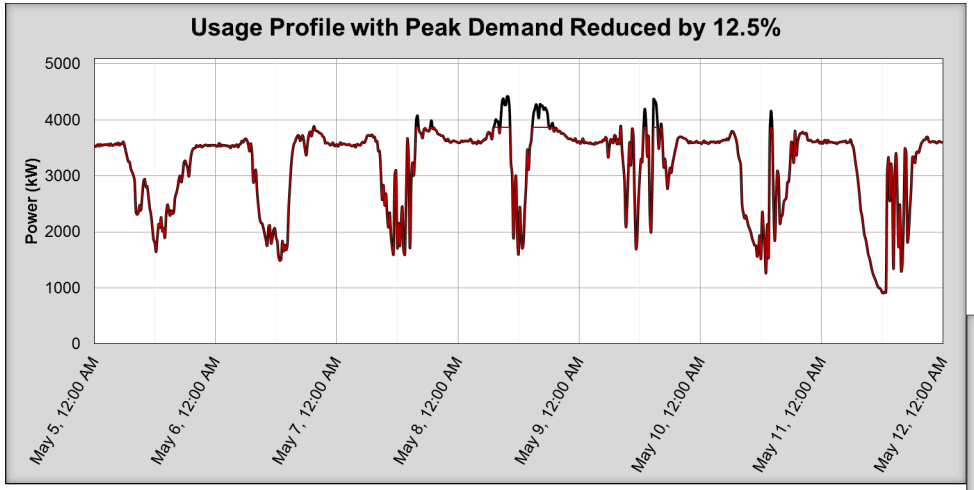
- Credit: Ryan Grant, Sandia National Labs, Preliminary; do not redistribute
- Study: Infiniband performance as processor power state reduced for on-node and offloaded protocol processing





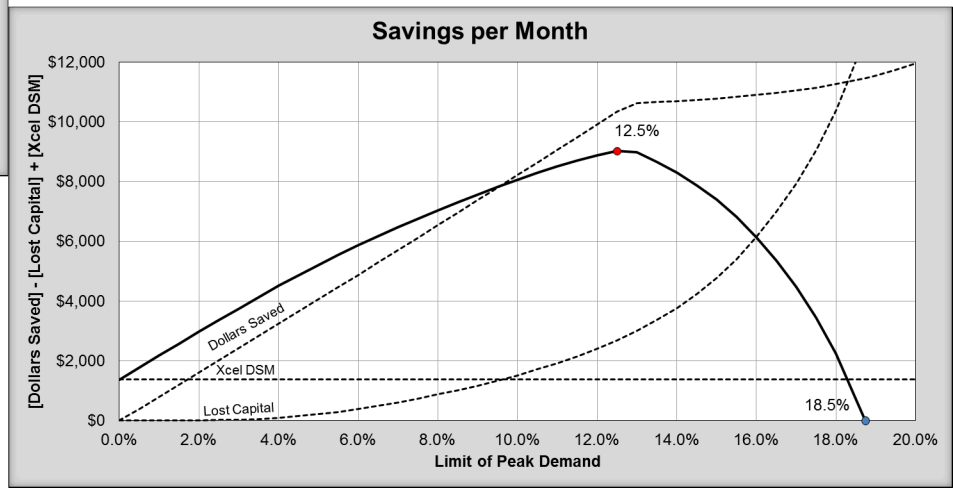
Actor: Facility Manager
System: HPCS Manager

- Credit: Ryan Elmore, National Renewable Energy Lab, Preliminary; do not redistribute
- Study: For electric bills based on peak usage periods, determine a maximum HPC load that minimizes loss of HPC use



⇦If HPC system can eliminate spikes by 12.5% (in black), the impact on HPC system is minimal

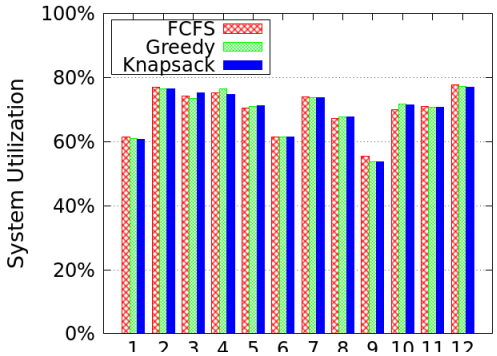
Flattening peaks more than 12.5% begins to significantly impact the capital investment in the system⇨

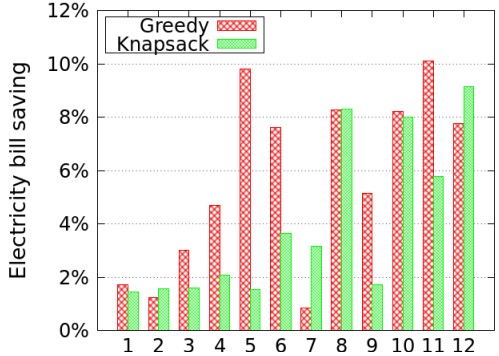


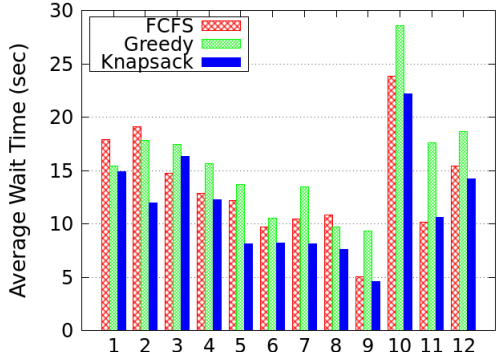
Actor: HPCS Manager
System: HPCS Resource Manager

- Credit: Xu Yang, Zhou Zhou, Sean Wallace, Zhiling Lan, Wei Tang, Susan Coghlan, and Michael E. Papka.Integrating dynamic pricing of electricity into energy aware scheduling for hpc systems. In Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, SC '13, pages 60:1–60:11, New York, NY, USA, 2013. ACM.
- Study: First Come First Served (FCFS) compared to Greedy and *0-1* Knapsack scheduling algorithms tuned for high-power consumption during lower priced electricity window and vice-versa

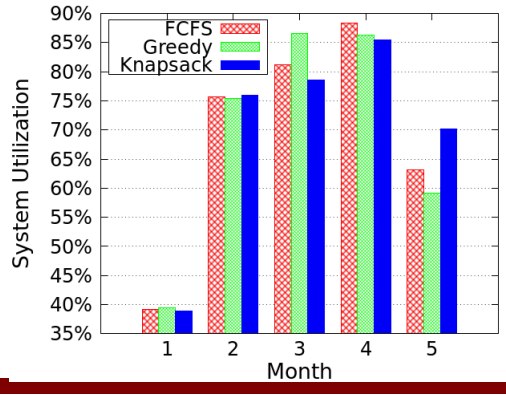
SDSC-BLUE

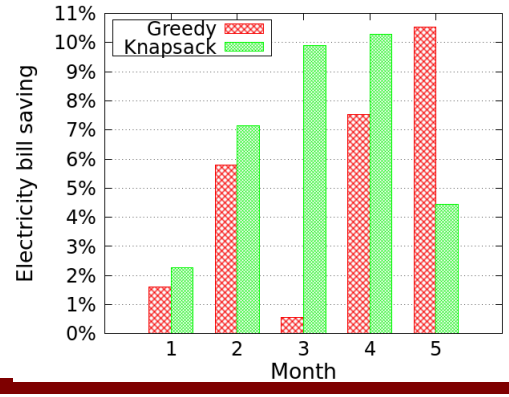


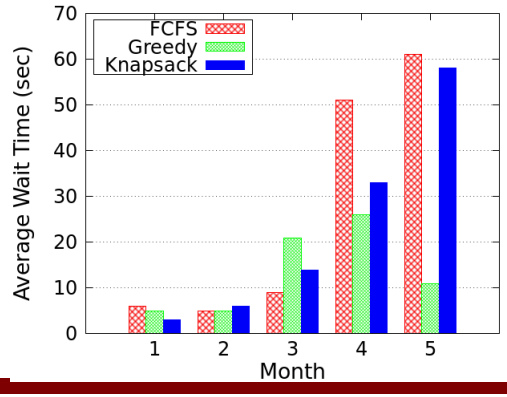




ANL-BGP



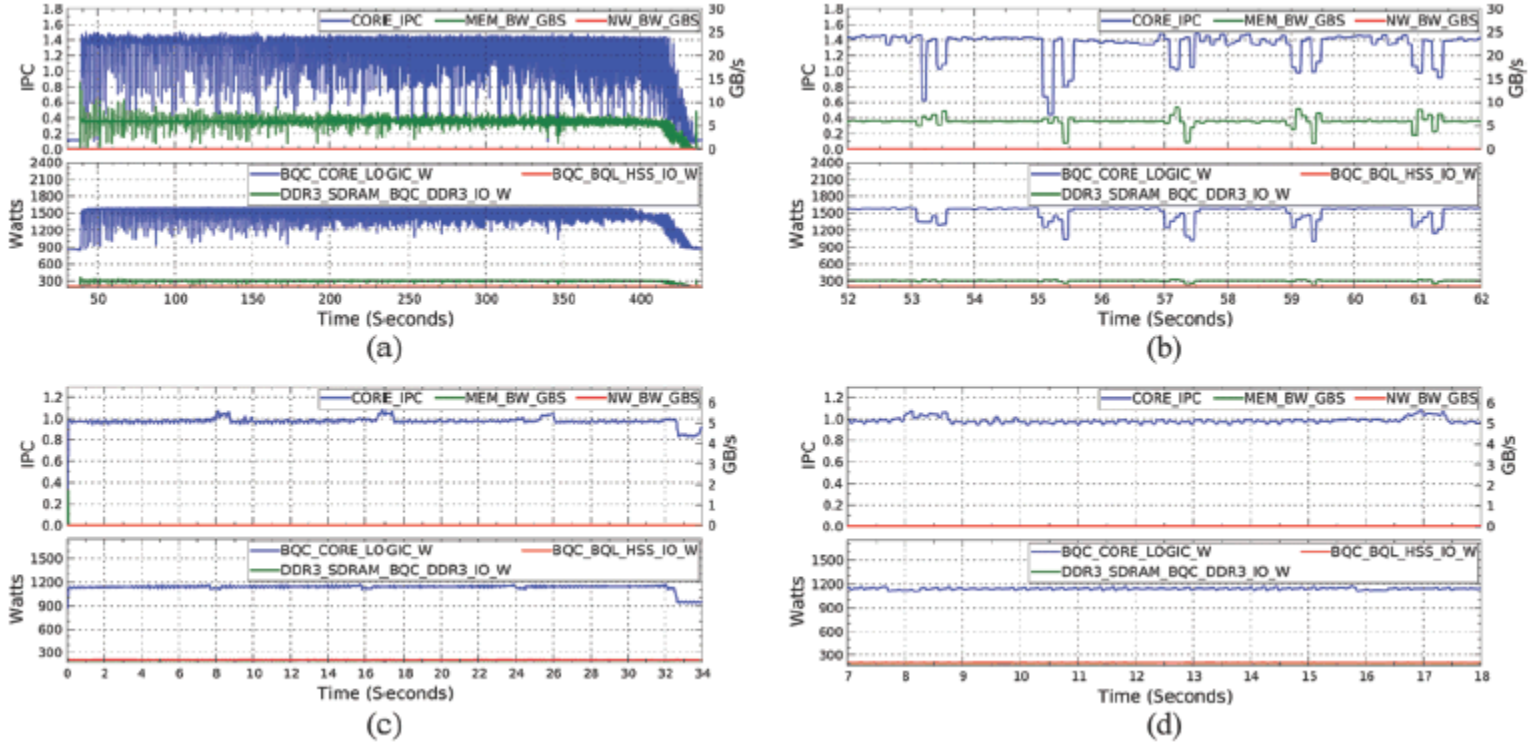




Utilization | Electricity Savings | Job Wait Time

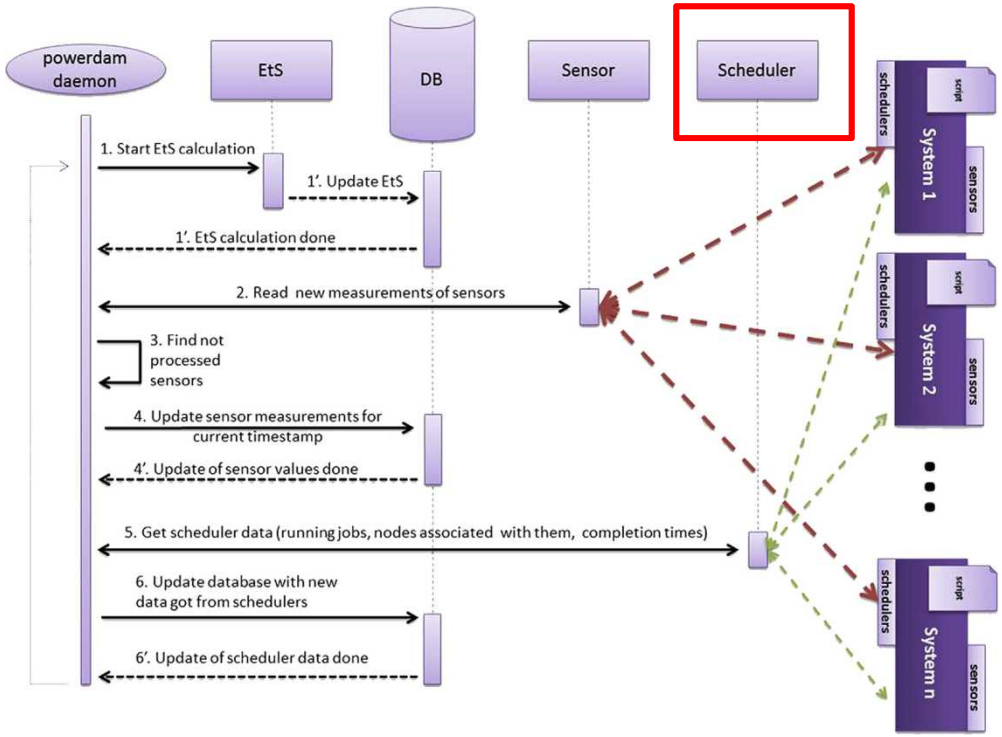
Actor: HPCS User
System: HPCS Resource Manager

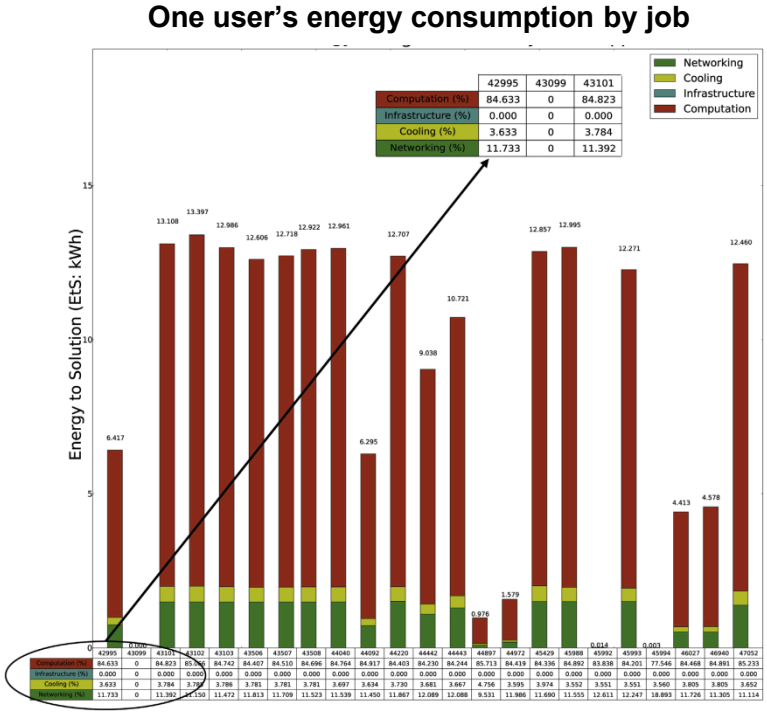
- Credit: R. Bertran, Y. Sugawara, H. M. Jacobson, A. Buyuktosunoglu, and P. Bose. Application-level power and performance characterization and optimization on IBM Blue Gene/Q systems. In IBM Journal of Research and Development, volume 57, 2013.
- Study: Once an application's power signature is analyzed, future job submissions can give power hints to the resource manager.



Actor: HPCS Resource Manager
System: HPCS Monitoring and Control

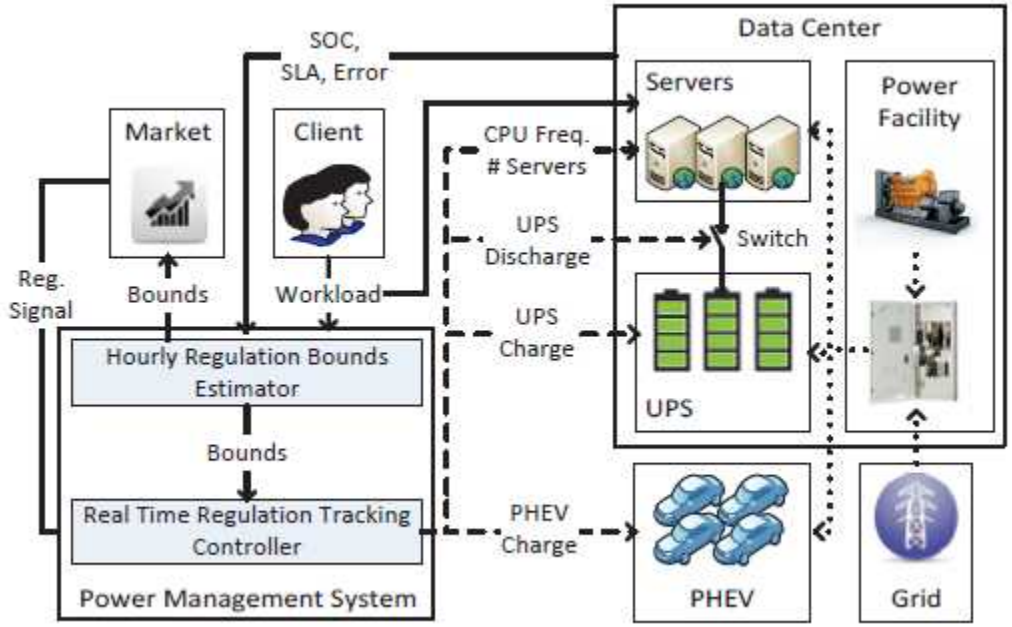
- Credit: Hayk Shoukourian, Torsten Wilde, Axel Auweter, and Arndt Bode. Monitoring power data: A first step towards a unified energy efficiency evaluation toolset for HPC data centers. Environmental Modelling Software, 2013.http://dx.doi.org/10.1016/j.envsoft.2013.11.011
- Study: Collection of power information and analysis





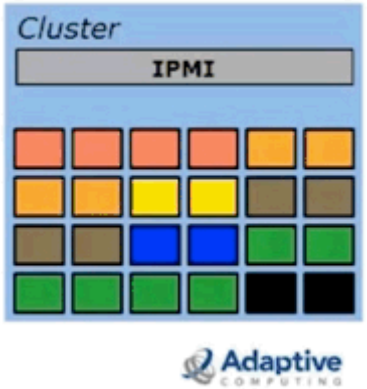
Actor: Facility Manager
System: Facility Hardware

- Credit: M. Brocanelli, Sen Li, Xiaorui Wang, and Wei Zhang. Joint management of data centers and electric vehicles for maximized regulation profits. In Green Computing Conference (IGCC), 2013 International, pages 1–10, June 2013.
- Study: Power Management system decides the power budgets for the servers and UPS in the data center, as well as electric vehicles



Actor: HPCS Resource Manager
System: HPCS Operating System

- Credit: Tiffany Trader. Green power management deep dive. Green Computing Report, June 2013. http://www.greencomputingreport.com/gcr/2013-06-05/green_power_management_deep_dive.html
- Study: Moab's Auto Power Management capabilities are front runners in introducing power as a resource to be allocated/managed.



Summary & Conclusions

Sandia National Laboratories

- Interesting HPC power research being done
- Portable APIs needed for general adoption and productization of the research
- Good mapping from research to our Power API interfaces

Additional Slides

- Hoping to get this far in 25 minutes

Actor: HPCS Operating System
System: HPCS Hardware

- Credit: V.M. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore. Measuring energy and power with papi. In Parallel Processing Workshops (ICPPW), 2012 41st International Conference on, pages 262–268, Sept 2012.
- Study: Experiences with writing drivers for PAPI's new power interfaces using existing internal and external power measurement components (e.g. RAPL and PowerMon)

Actor: HPC Accounting (and HPCS User) System: HPCS Monitoring and Control

Sandia National Laboratories

- Credit: Yiannis Georgiou. Energy accounting and control on HPC clusters, November 2013. http://perso.ens-lyon.fr/laurent.lefevre/greendayslille/greendayslille_Yiannis_Georgiou.pdf
- Study of ways to monitor and measure energy consumption

SWF trace with Consumed Energy field

```
JobID Submit Wait Elapsed CPUs CPUTime Mem ...ConsumedEnergy
3541439 0 271 67 1 59 374756 ..........................  20960
3541440 5 266 50 1 41 356628 ..........................  12150
```

Actor: HPCS Admin System: HPCS Monitoring and Control

- Credit:
- Study:

Actor: HPCS Monitor and Control System: HPCS Hardware

Sandia National Laboratories

- Credit:
- Study: