The Second Workshop on HPC Power Management:
Knowledge Discovery

*Power API Collaborations, Community, and What's Next*

25 August 2016

James H. Laros III
Principal Member of Technical Staff
Sandia National Laboratories
http://powerapi.sandia.gov
If I had an hour to solve a problem and my life depended on it, I would use the first 55 minutes determining the proper questions to ask.

Albert Einstein
2004-2006: Initial Research

Measure

Control

- HPCS Monitor & Control
- HPCS User
- Facility Manager
- HPCS Manager
- HPCS Operating System
- HPCS Application
- HPCS Admin
- HPCS Resource Manager
- HPCS Accounting
Diagram is the result of a UML study of the target space
• Goal: Define Scope, Roles and Interfaces

Arrows indicate interfaces or interaction between an Actor (Role) and System
• Each interaction represents an interface that is defined in the specification
• Specification is structured from the user or Role perspective

Notice that an Actor (Role) can also be a System

2014: Initial Version Power API Specification

- Versions 1.0, 1.1, 1.1a, 1.2 and 1.3 delivered
- Community needed a portable API for measuring and controlling power and energy
- Sandia developed Power API specification to fill this gap
- Provides measurement and control interfaces designed to enable portability
  - Covers full spectrum of facility to component
- First production implementation will be Trinity (ATS1)
- Continued (increasing) community involvement and influence
  - This is what we are here to promote!

http://powerapi.sandia.gov

- **Broad Scope**
  - High-level: end user and applications
  - Low-level: hardware and operating system

- **Roles (actors)**

- **Systems**

- **Interfaces**
  - Roles interacting with Systems
Power API Goals

- Portability for the HPC community
  - Wouldn’t it be nice to develop tools that worked on all your machines with little to no modification?
  - Same desire exists no matter what Role you play

- Forecast emerging needs of HPC community
  - As a group, inform the vendors of how we want to use systems now and in the future
  - Specification acts as a basis of collaboration

- Expose new capabilities developed by vendors and community
  - Leverage vendor and community innovations in this and related spaces
  - E.g. Geo and Redfish

- Most important, want something out there to throw stones at
  - Need a starting point!
What is the Power API?

- A comprehensive API for power MEASUREMENT and CONTROL of HPC platforms
  - Comprehensive = Facility to Component
  - API = Define the interface not the mechanism
  - HPC platforms = Facility (or datacenter) and all the platforms within

- Core (Common) among all “users” Includes:
  - Roles, Initialization, Navigation, Objects and Groups,
  - Attributes (Get/Set), Metadata and Statistics

- High-Level Common
  - Higher level of abstraction but still potentially common among multiple Roles

- Role/System Specific
  - Higher level abstraction specific to how Role interfaces with system
So what have we been doing since last year?

- Two new versions released this year with a big one coming
  - Versions 1.2 and 1.3 largely changes resulting from NRE collaborations
  - Version 1.4 possible
  - Version 2.0 will include Python bindings
- 2\textsuperscript{nd} BoF at SC15 – standing room only!
- Hopefully a 3\textsuperscript{rd} at SC16
  - Focus on how to move to a more community influenced/driven paradigm
  - We want your great ideas now and at the BoF!!
- RNET
  - Efficient large scale sampling
  - Reporting power along side performance data
Intel
- Working closely with the goal of compatibility between GEO and Power API
- API focus has been our Application->OS interface
  - Since theirs is largely a runtime effort
- In particular the AppTuningHint() interface
  - Multiple nested phases (Version 1.4 or 2.0)

Continuing to work with Cray Inc. (Trinity NRE)
- Cray’s Power Management Data Base (PMDB)
  - Python implementation of Power API (Version 2.0)
- Compute Node Interface
  - C implementation of Power API

Began work with Adaptive Computing (Trinity NRE)
- Power aware scheduling use cases for Trinity
- Exercise portions of Cray Power API implementation

We would like to involve more HPC community members in driving these (and new) efforts forward
Python Implementation of Power API

```python
>>> cntxt = pwr.Cntxt(pwr.Role.ACC, "System Accounting")
>>> entryPoint = cntxt.GetEntryPoint()
>>> entryPoint.GetName()
's0'

>>> entryPoint.AttrGetValue(pwr.AttrName.POWER)
InfoFromGet(attr=6, value=26839, obj=<cray.obj.ObjPlatform object at 0x7f4fc3805ed0>,
timestamp=1471899692.880717, rc=0)

>>> entryPoint.AttrGetValue(pwr.AttrName.POWER).value
26839

>>> entryPoint.AttrGetValue(pwr.AttrName.ENERGY)
InfoFromGet(attr=12, value=139260873, obj=<cray.obj.ObjPlatform object at 0x7f4fc3805ed0>,
timestamp=1471899709.188973, rc=0)

>>> entryPoint.AttrGetValue(pwr.AttrName.ENERGY).value
139325195
```
>>> now = time.time()
>>> tempTimePeriod = pwr.TimePeriod(now - 300.0, now)
>>> myNode = cntxt.GetObjByName("c0-0c2s12n0")
Who is Behind PowerAPI?
Wish List

“My job” defined as: HPC User, Administrator, Application, etc.

- A standard way of interfacing with HPC systems to measure and control power!
- Adoption of this (aforementioned) standard by the HPC vendor community
- A growing set of tools that use these standard interfaces (portable)
- Integration of app libraries and runtimes that utilize these interfaces
- Minimum standards for measurement and control
  - Node, component, etc.
  - Sample frequency
  - Quality of sample
  - Time-stamp accuracy
Questions?

- Register on the reflector
- Get the current version of the spec
- Get the prototype/reference implementation source
- Other information as it develops
- Please get involved and help us (the community) improve the specification
- Sandia TEAM:
  James Laros, Suzanne Kelly, Kevin Pedretti, Michael Levenhagen, Ryan Grant, Stephen Olivier