Metadata for System Health in HPC

Line Pouchard

July 13, 2010

Reliability, Availability, Serviceability (RAS)

- What is RAS?
  - An effort to avoid, predict, mitigate or remedy faults that cause applications to terminate unsuccessfully.
- Why is it important?
  - As clusters increase in size and complexity, the latency of performing check-points/restarts will exceed MTTF and MTTI.
  - Failures impact cost of operations.
  - Usage profiles can be used to provide application signatures.
- Numerous issues hinder progress
  - No standard practices for data collection (what is collected and how) and data representation.
  - Root-cause analysis for failure attribution not performed. Failures may be due to HW, SW, OS, libraries, applications, or a combination of these.
  - Numerous methods for failure prediction but only one large-scale collection of operational data available to public scrutiny.
  - Data collection itself can have an impact on operations, so it may be turned off when strain on the system increases.

This talk covers

- Data collection
- Preliminary Analysis

This talk does not cover

- Methods for improving check-pointing
- OS virtualization

Why and what to collect for system health data?

- Power Consumption
- Improve resource management and scheduling
- Detect application signatures (power, intrusion detection)
- RAS (Reliability, Availability, Serviceability)
- Temperatures at core, node, cabinet, bridge, router, VRMs
- Voltages at same
- Energy
- Job and user Ids
- What runs where when and for whom

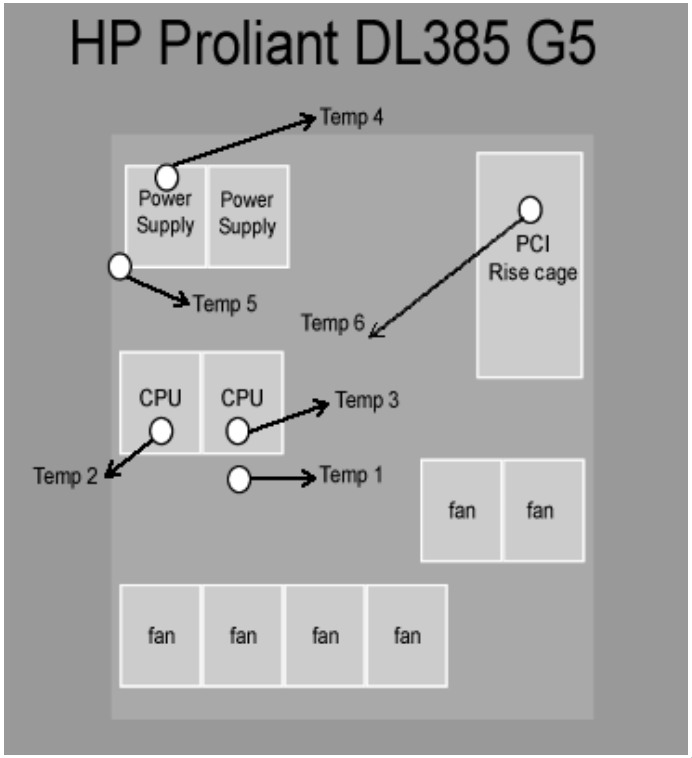
Data Collection and Metadata Challenges

- Massive amount of data streaming from potentially thousands of nodes
  - Zillions of small files/messages.
  - Because of volume, data is analyzed on the fly, only the results are persisted, precluding forensic analysis.
- Heterogeneous sensors and sources
  - Sensor data variables not always accessible.
  - Many forms of instrumentation with output in multiple formats
  - Reliance on proprietary instrumentation with no verification and validation.
  - Calibration is sometimes ignored.
  - Sensor output embedded in (proprietary) system monitoring tools like CRMS.
- System sensor data is represented with numerous schemas and data models
  - Standards exist but no single standard contains all the needed elements.
- Challenges in data result in contradictory results in the literature.

Collected Data

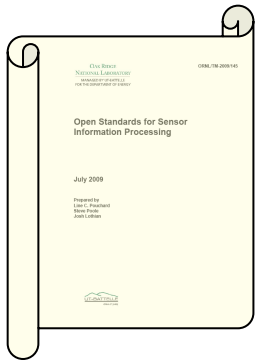
- Internal collection
  - Temperatures (6)
    - system board near heatsink
    - CPU sockets 1 & 2
    - power supply cage,
    - power supply riser connector,
    - back edge of board
  - Fan speeds (2)
  - No uasable voltages
  - Cpu
    - Cpu usage, idle time, averages per node, node uptime.
  - Memory free, cached, active.
- External collection
  - 4.9 million temperatures
  - Power consumption in progress

- 8.5 Gb total amount of sensor data for Trident
  - 126.0 million temperatures
  - 90.6 million cpu speed metrics
  - 11.3 million memory usage metrics



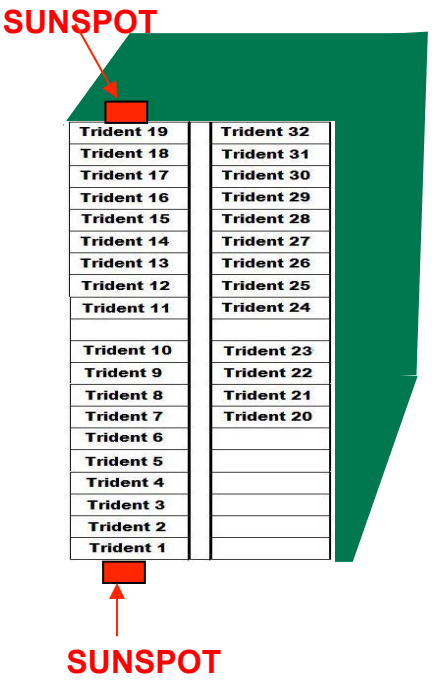
Heterogeneous data sources and model representation

- IEEE 1451
  - Standard for a Smart Transducer Interface for Sensors and Actuators
- Open-Geospatial Consortium
- Lm-sensors
- IPMI sensors on the testbed
  - The IPMI source code is not available.
  - IPMI operates at the BIOS level.
  - Mapping measurements to sensor in a node is a matter of guesswork.
- Findings:
  - No single specification/package satisfies the goal of providing a model suitable to analyze sensor data from all manufacturers. Some can be adapted.



Open Source Standards for Sensor Information Processing: ORNL/TM-2009/145 (Pouchard, Poole, Groer, Lothian)

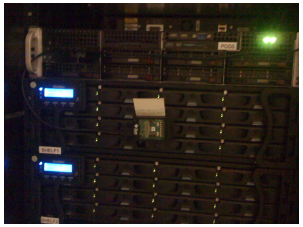
Instrumenting a testbed Cluster



- 32 nodes, 2 quad core AMD processors, OEM: HP
- Internal data sources:
  - Ipmi
  - Memory
  - Cpu usage
- External data sources
  - 13 Sun Small Programmable Object Technology.
  - ServerTech Sentry Switched CDU.

With Josh Lothian and Chris Groer

Sun™ Programmable Object Technology (SunSPOTs)



Cabinet	Location
BV4	Front of POD5
B011	Panasas disks
CB04	Top of Trident cabinet
BU4	Switch PID6
BV4	PID7
BW4	Anue 4 - Network emulator
BY03	IBM -
BZ04	Top of POD14 - switch
CA04	Trident-mgmt2
CB04	Trident32
Cray XMT	Left side – powered by trident18
CC04	Top of cabinet – powered by Trident18
CC04	Trident1

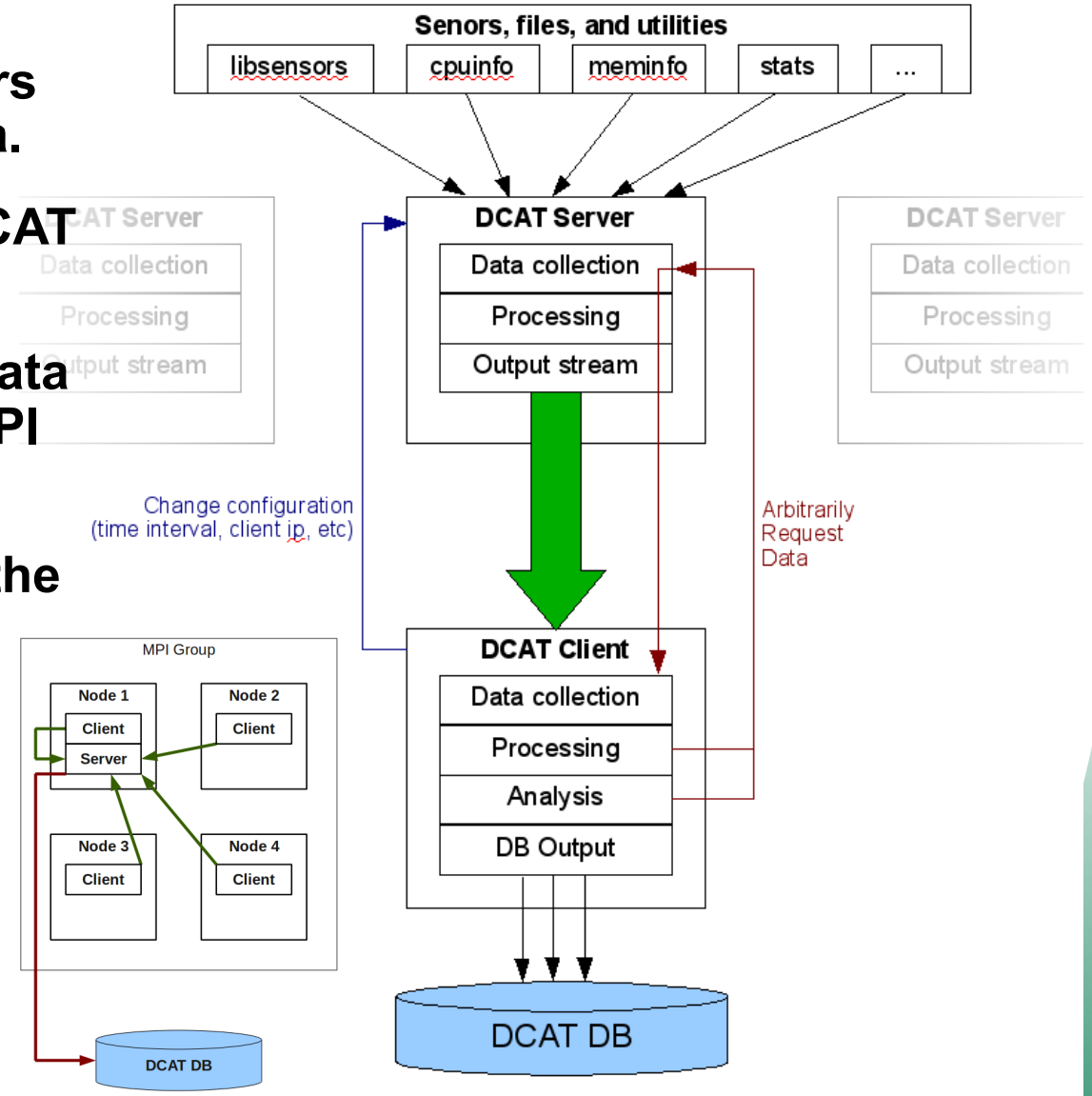
- **41x23x70 mm, 54gr., range varies depending on interference.**
- **Powered by USB.**
- **Contains an Accelerometer, a light sensor, and a temperature reader.**
- **Communicates with a base-station (same size just a little thinner) with its own IEEE 802.15.4 wireless network.**
- **Uses JRE and Ant technology (java based).**
- **Support authentication with certificates.**
- **Unauthorized over-the-air deployment is precluded.**
- **Ownership changes only allowed with physical access.**
- **Data is encrypted.**
- **We collect all data (on a 32-bit system).**

ServerTech Switched Cabinet PDU: Power Management

- ServerTech Switched Power Distribution Unit (PDU)
  - 24 power outlets per Cabinet (CDU)
  - Per outlet power sensing (POPS)
    - Measure current, voltage, power, …
    - accuracy of ± 2%
  - Installed on Trident (4 PDUs for two racks)
  - Various communication protocols
    - SNMP, ssh, serial, ….

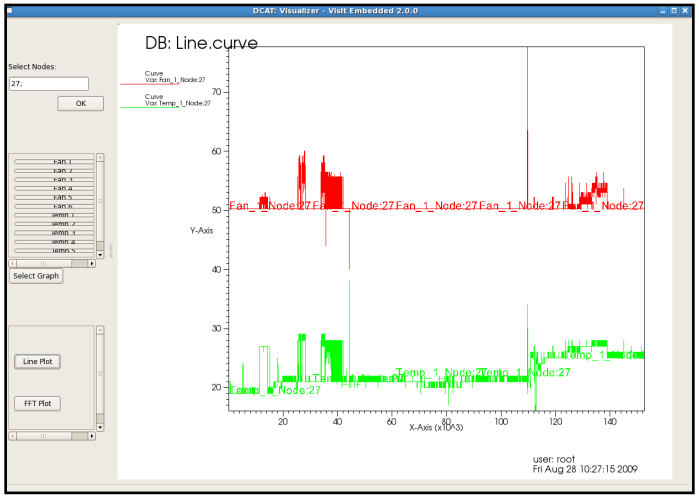
DCAT: a RAS data collection tool

- Each DCAT node gathers system and sensor data.
- The data is stored in DCAT as C structs.
- The node passes that data to the client node via MPI as derived datatypes.
- The client node stores the data in its database.

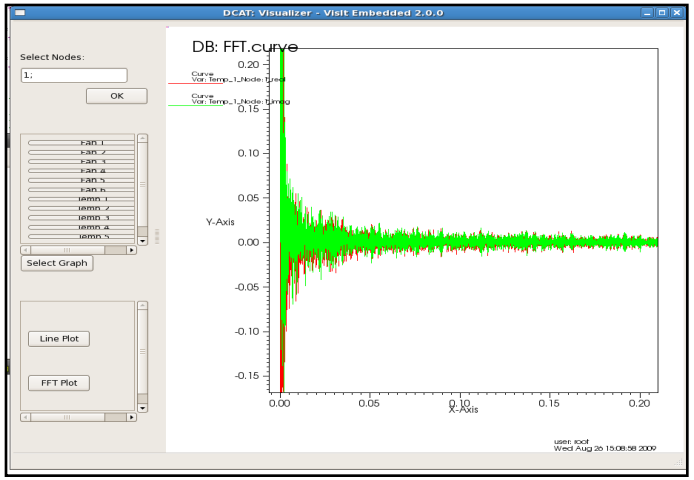


Visualization

- The Vis. plug-in uses the VisIt Engine
  - VisIt is a parallel visualization engine and environment developed at Sandia and ORNL.
- The Plug-in enables an API to VisIt.
- Includes a User Interface.
- FFT algorithm applied to temperature data
- Node 3: Temp 1, 1024 readings, CPU socket 2.

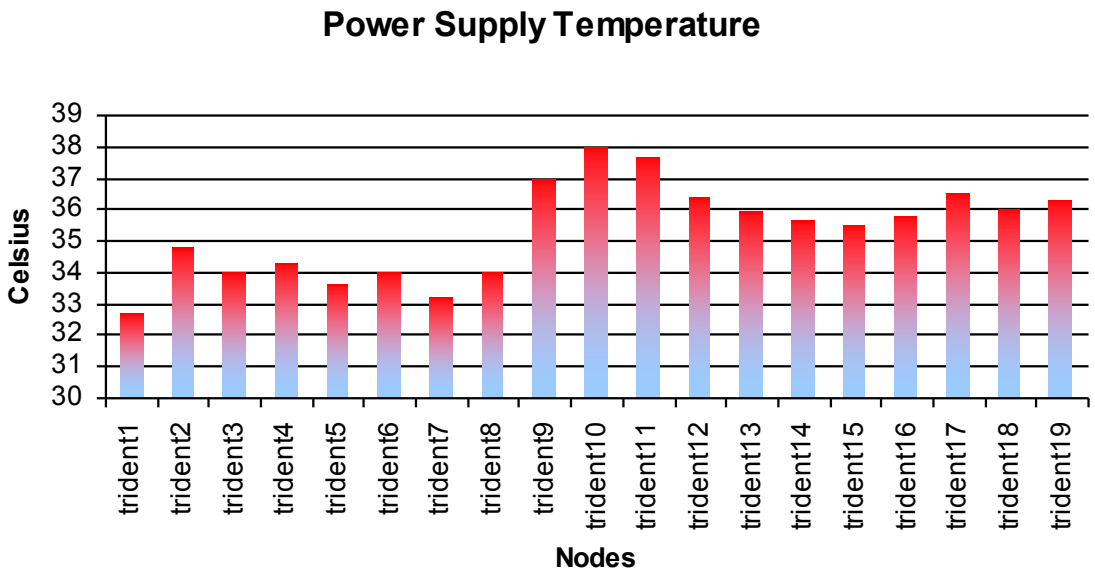


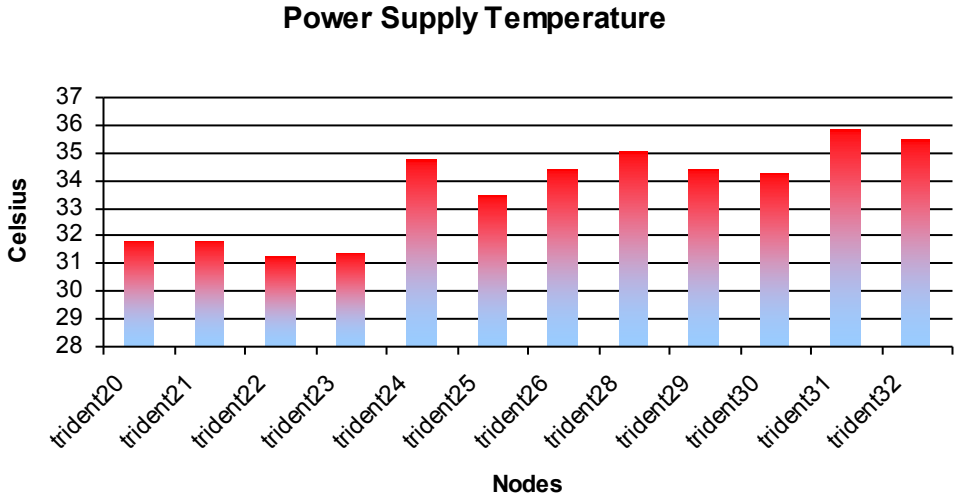
Temperature and Fan Data for Trident 1



Temperature Spectrum Data for Trident 1

Analysis at idle



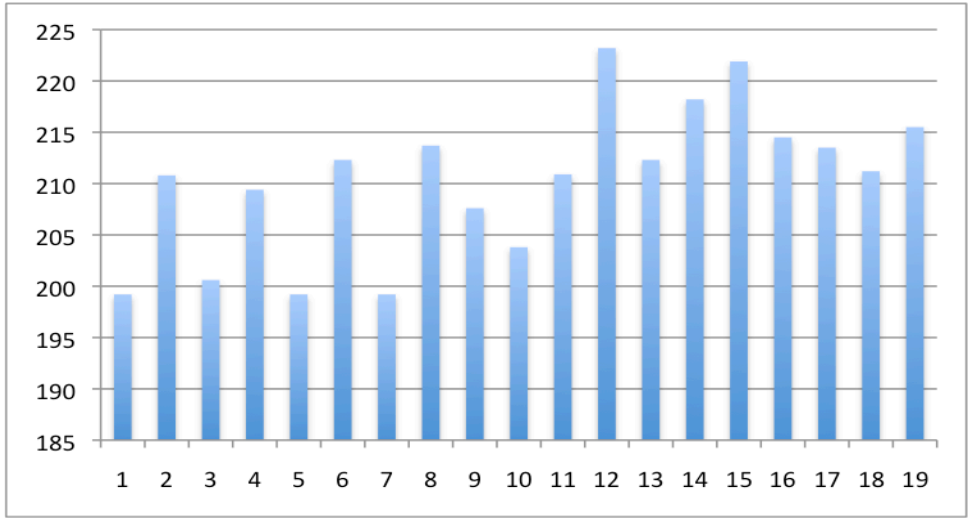


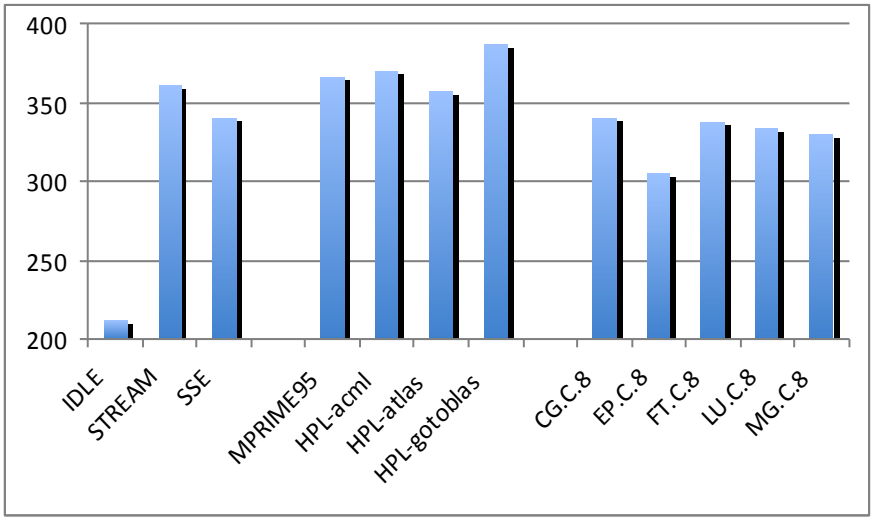
Trident 19	Trident 32
Trident 18	Trident 31
Trident 17	Trident 30
Trident 16	Trident 29
Trident 15	Trident 28
Trident 14	Trident 27
Trident 13	Trident 26
Trident 12	Trident 25
Trident 11	Trident 24
Trident 10	Trident 23
Trident 9	Trident 22
Trident 8	Trident 21
Trident 7	Trident 20
Trident 6	
Trident 5	
Trident 4	
Trident 3	
Trident 2	
Trident 1	

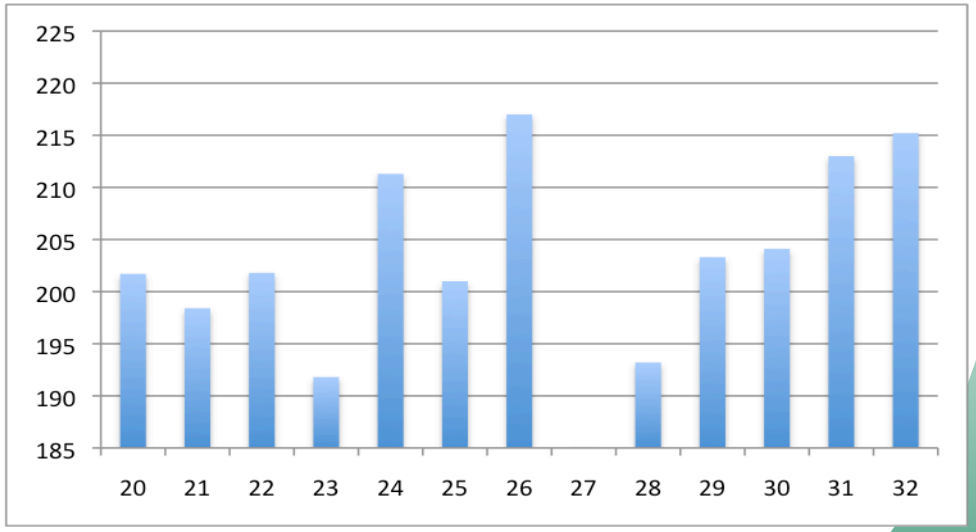
- Nodes are heterogeneous for temperatures and power consumption
- However, thermal profiles and power profiles also differ
- Trident 9, 10, and 11 are the hottest
- Trident 12 & 14 consume the most power

Power Management

- **Heterogeneity in power consumption per applications**
  - **trident 16 uses a higher voltage by default (see running HPL)**







Source: Idle for 10 minutes. X: trident node, Y: average watts

Thank You!

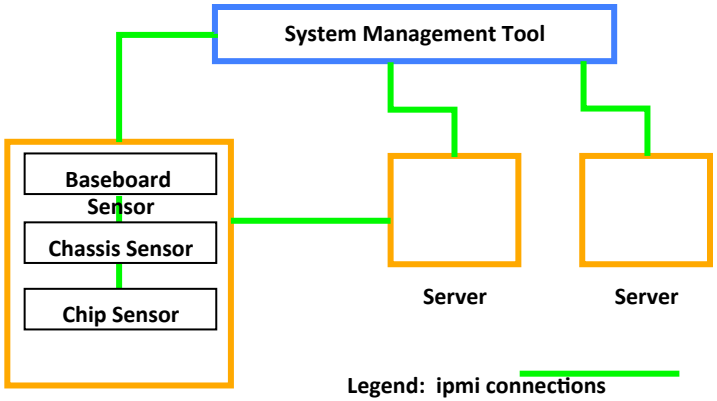
Acknowledgments:

This work was supported by the United States Department of Defense and used resources of the Extreme Scale Systems Center at Oak Ridge National Laboratory.

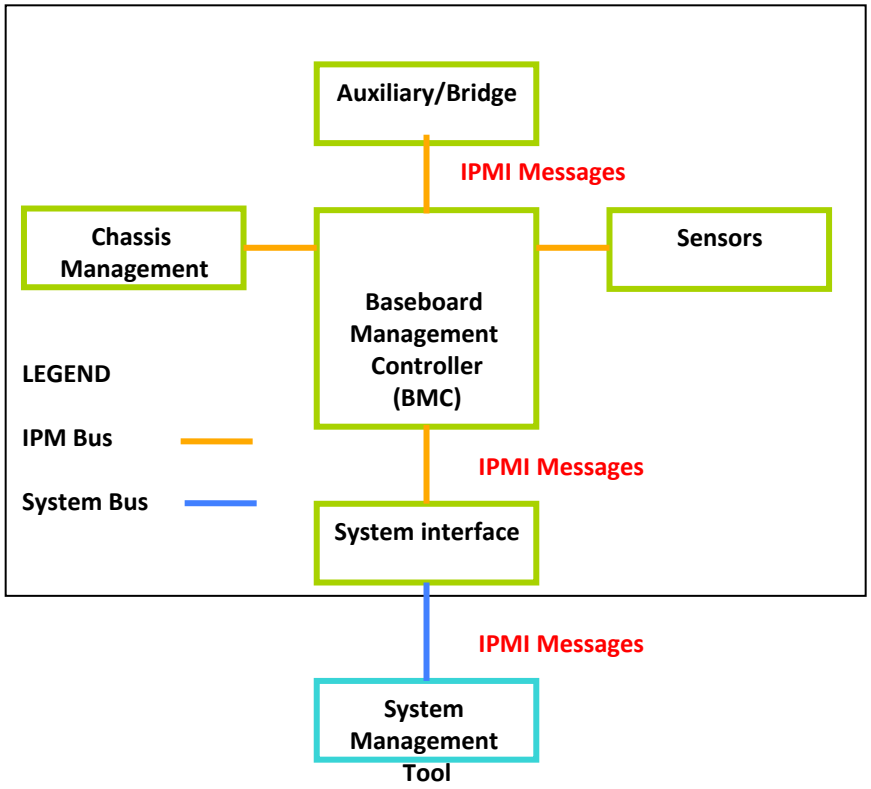
Additional Slides

Ipmi – Intelligent Platform Management Interface

IPMI in a managed system



IPMI logical architecture



Promoters:

- Dell
- HP
- Intel Corporation
- NEC Corporation

Not supported by:

- Cray

Adopted by:

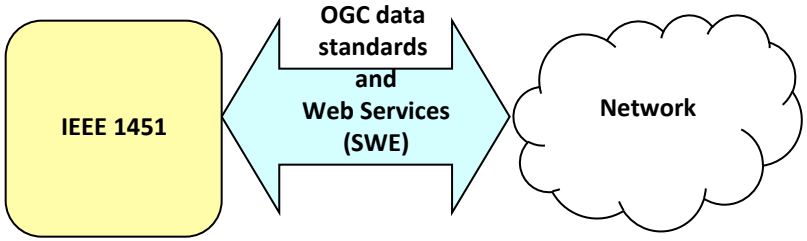
- >200 OEMs

- **The Specification is Open Source (latest v.2.0)**
- **Tool implementation takes place at the BIOS level**
- **Requires proprietary drivers**

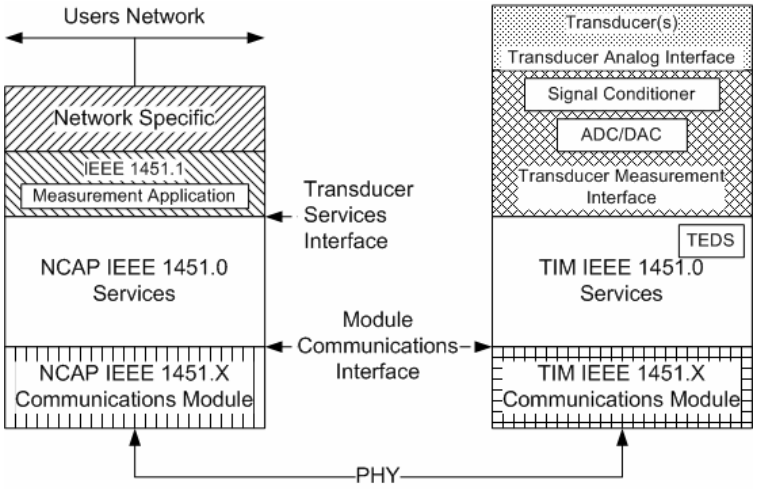
IPMI – Intelligent Platform Management Interface Specification

- Specification initially created by Intel
- Promoted by Dell, HP, Intel, NEC, adopted by more than 200 OEMs (not Cray)
- Open source specification, requires drivers and proprietary implementations
- Sensor data is queried via *ipmitool* –job outputs data once per minute

IEEE 1451: Transducer Electronic Data Sheet (TEDS)



OGC SWE standards serve as an interface between IEEE 1451 and a network.



- **Data structure of a TEDS**
  - **Unsigned integer 32, 4 octets**
  - **MetaTEDS (internal timeout value)**
  - **Transducer Channel (sensor metadata)**
  - **User's Transducer Name**
  - **Frequency response**
  - **Calibration**
  - **Transfer function**
  - **Command (sensor control)**
  - **Geo-location**
- **Represents a sensor, not the data transmitted by a sensor**

TIM: Transducer Interface Model

NCAP: Network Capable Application Processor

PHY: Physical Connections

ADC: Analog-to-Digital Conversion

DAC: Digital-to-Analog Conversion

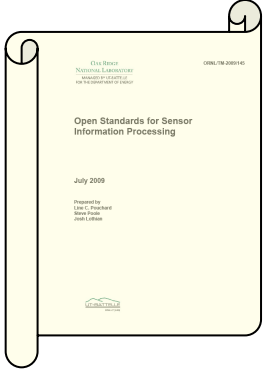
False Alarms

- Trident is running HP Systems Insight Manager – uses ILO (Integrated Lights Out) system which has its own management processor
- Hundreds of alerts : “The server's temperature is outside of normal operating range. The system will be shutdown”
- However, reviewing our IPMI logs indicates nothing unusual

RAS in FY 2010

- Research
  - Literature survey of RAS in cloud computing
  - Taxonomy of faults and failure events based on DMTF's Common Information Model
  - Testing various failure prediction methods
  - Power management
- Development
  - Reliable RAS architecture
  - Collecting data on Cray SEDC (internal and external)

Publications



- Pouchard, Poole, Lothian, Groer, "Open Source Standards for Sensor Information Processing," ORNL/TM-2009/145
- Chung-Hsing Hsu and S. Poole "An Adaptive Run-Time System for Improving Energy Efficiency," in Green Computing: Large-Scale Energy Efficiency, W. Feng, ed, Chapman & Hall / CRC Press, 2010.
- Pouchard, Dobson, Poole, "Collecting Sensor Data for High-Performance Computing: A Case-study,"Parallel and Distributed Processing Techniques and Applications (PDPTA), WORLDCOMP 10, Las Vegas, July 12-15, 2010.