Data Collection and Analysis for the Health of Systems

Dr. Line Pouchard
Jonathan Dobson
Extreme Scale Computing Systems
Dr. Stephen Poole, Director

Polytechnic University of Puerto Rico, December 14, 2009

Managed by UT-Battelle for the U. S. Department of Energy

OAK RIDGE National Laboratory

OUTLINE

- The problem
- Existing utilities
- Data Representation Standards
- Data Collection at ORNL. Our tool: DCAT
- Visualization
- Data Validation: SunSPOTS
- Future work

Data Collection and Analysis for System Health

Problem:

As HPC clusters increase in size and complexity, it becomes increasingly difficult to locate errors and pinpoint failures.

Objective:

This effort aims to provide a data collection architecture and analysis tools for the analysis of large, composite data sets in an HPC environment.

Method:

By analyzing sensor information and other health-related parameters, locating and predicting faults becomes more efficient.

Why and how?

- What kind of information do we want and/or can we get out of machines about themselves?
  - Temperatures, voltages, fan rpms, power supplies
  - Sometimes acoustic data, air flow
  - Memory-related statistics
- Why:
  - Failure prediction
  - Performance measurements
  - Anomaly detection
  - Load balancing
  - Memory usage
- From which components?
  - All the components that impact the above tasks
  - From components for which this information is available
  - New operating systems will need to provide this data
- How?
  - From sensors embedded in the components or on the boards by manufacturers
  - Some general open source software, utilities and drivers
  - Other manufacturer-dependent libraries

Example: Jaguar

- The XT4 partition contains 7,832 compute nodes in addition to dedicated login/service nodes. Each compute node contains a quad-core AMD Opteron 1354 (Budapest) processor running at 2.1 GHz, 8 GB of DDR2-800 memory (some nodes use DDR2-667 memory), and a SeaStar2 router. The resulting partition contains 31,328 processing cores, more than 62 TB of memory, over 600 TB of disk space, and a peak performance of 263 teraflop/s (263 trillion floating point operations per second).
- The XT5 partition contains 224,256 compute cores in addition to dedicated login/service nodes. Each compute node contains two hex-core AMD Opteron processors, 16GB memory, and a SeaStar 2+ router.

What are the Data Problems?

- Massive amount of data streaming from potentially thousands of nodes
  - Zillions of small files/messages.
- Heterogeneous sensors and data representation
  - Accessible sensor data variables
  - Measurement outputs in numerous formats
  - Proprietary schemas
- Issues:
  - Requesting data from the sensors (push, pull, poll)
  - Metadata and Interoperability of data models for processing tools
  - Fusing data for analytics
  - Control of sensor operations

Computer system sensor data is represented with numerous schemas and data models

Standards exist but no single standard contains all the elements we need

Mixing and matching elements will require a significant effort (syntax, semantics, hierarchies)

Data representation

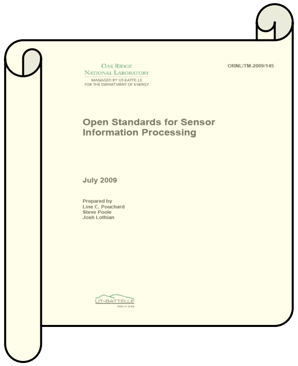
- Standardized data models
  - IEEE 1451
  - Open Geo-Spatial Consortium
    - SensorML
    - TransducerML
    - Operations and Measurements
  - Common Instrument Middleware Architecture (CIMA)
- Various data output formats from utilities
  - lm-sensors
  - Hardware Monitoring for Mac OSX
- Computer Information Management
  - Distributed Management Task Force
    - dmidecode
    - Common Information Model (CIM)

Outline

- **Existing Utilities and Standards**

Open Source Standards for Sensor Information Processing: ORNL/TM-2009/145 (Pouchard et al.)

- IEEE 1451
  - Standard for a Smart Transducer Interface for Sensors and Actuators
  - Open-Geospatial Consortium
- Lm-sensors
- IPMI and Trident sensors
  - Adapters exist for lm-sensors and IPMI.
  - The IPMI source code is not available.
  - IPMI operates at the BIOS level.
  - Identifying which sensor reports which measurement 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.



Linux monitoring utility (lm-sensors)

- An open source, hardware-monitoring effort measuring and reporting the health of systems.
- Sensor detection and common libraries
  - Sensor detect
  - Sensor lib
- Additional drivers depending on hardware specification
- Records core and other temperatures, voltages, and fan speeds.

Lm-sensors on pid1 and pod1 testbed

```
w83627hf-isa-0290
Adapter: ISA adapter
VCore 1:   +3.90 V  (min =  +1.34 V, max =  +1.49 V)       ALARM
VCore 2:   +3.79 V  (min =  +1.34 V, max =  +1.49 V)       ALARM
+3.3V:     +3.82 V  (min =  +3.14 V, max =  +3.46 V)       ALARM
+5V:       +5.27 V  (min =  +4.73 V, max =  +5.24 V)       ALARM
+12V:     +11.67 V  (min = +10.82 V, max = +13.19 V)
-12V:      +0.88 V  (min = -13.18 V, max = -10.88 V)       ALARM
-5V:       +1.89 V  (min =  -5.25 V, max =  -4.75 V)       ALARM
V5SB:      +5.59 V  (min =  +4.73 V, max =  +5.24 V)       ALARM
VBat:      +0.08 V  (min =  +2.40 V, max =  +3.60 V)       ALARM
fan1:        0 RPM  (min =    0 RPM, div = 2)
fan2:        0 RPM  (min =    0 RPM, div = 2)
fan3:        0 RPM  (min =    0 RPM, div = 2)
temp1:       -48°C  (high =   -79°C, hyst =   +22°C)   sensor = thermistor
temp2:     -48.0°C  (high =   +80°C, hyst =   +75°C)   sensor = thermistor
temp3:     -48.0°C  (high =   +80°C, hyst =   +75°C)   sensor = thermistor
vid:      +1.419 V  (VRM Version 11.0)
alarms:
beep_enable:
          Sound alarm enabled
```

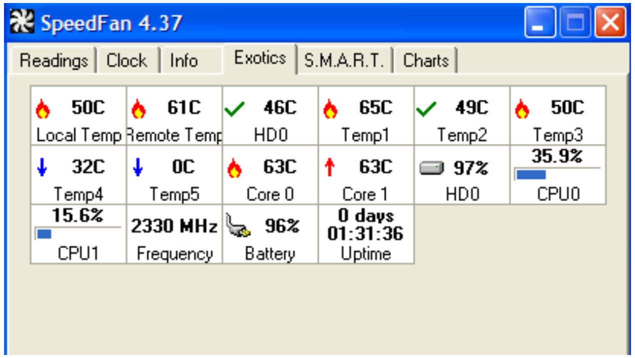
Intel Xeon

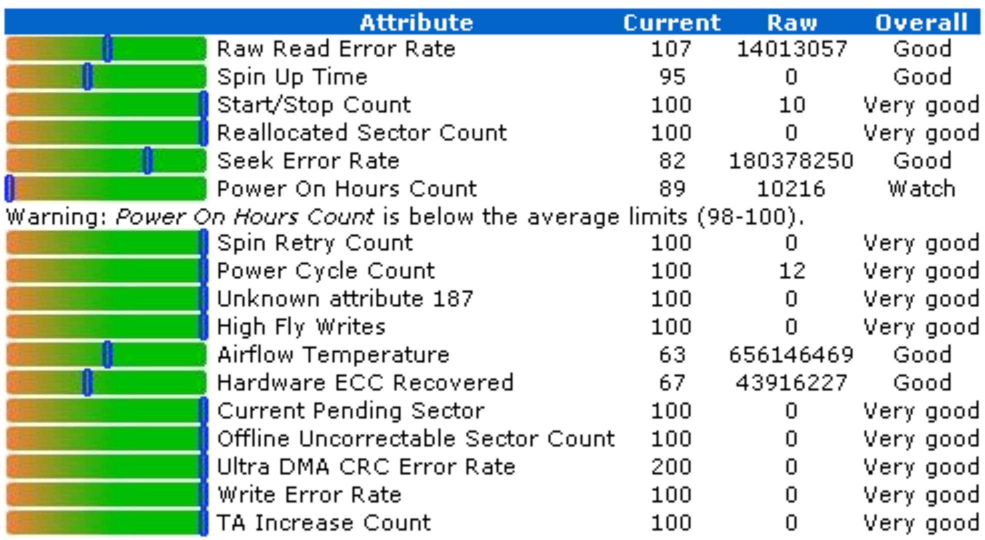
```
w83627hf-isa-0290
Adapter: ISA adapter
VCore 1:   +3.94 V  (min =  +0.00 V, max =  +0.00 V)       ALARM
VCore 2:   +3.89 V  (min =  +0.00 V, max =  +0.00 V)       ALARM
+3.3V:     +3.86 V  (min =  +3.14 V, max =  +3.47 V)       ALARM
+5V:       +5.03 V  (min =  +4.76 V, max =  +5.24 V)
+12V:     +11.61 V  (min = +10.82 V, max = +13.19 V)
-12V:      +0.80 V  (min = -13.18 V, max = -10.80 V)       ALARM
-5V:       +1.84 V  (min =  -5.25 V, max =  -4.75 V)       ALARM
V5SB:      +5.35 V  (min =  +4.76 V, max =  +5.24 V)       ALARM
VBat:      +0.64 V  (min =  +2.40 V, max =  +3.60 V)       ALARM
fan1:        0 RPM  (min =  712 RPM, div = 8)              ALARM
fan2:        0 RPM  (min =  712 RPM, div = 8)              ALARM
fan3:        0 RPM  (min = 5314 RPM, div = 2)              ALARM
temp1:       -48°C  (high =  +120°C, hyst =  +115°C)   sensor = thermistor
temp2:     -48.0°C  (high =   +80°C, hyst =   +75°C)   sensor = thermistor
temp3:     -48.0°C  (high =   +80°C, hyst =   +75°C)   sensor = thermistor
vid:      +0.000 V  (VRM Version 0.0)
alarms:
beep_enable:
          Sound alarm enabled
```

AMD Opteron

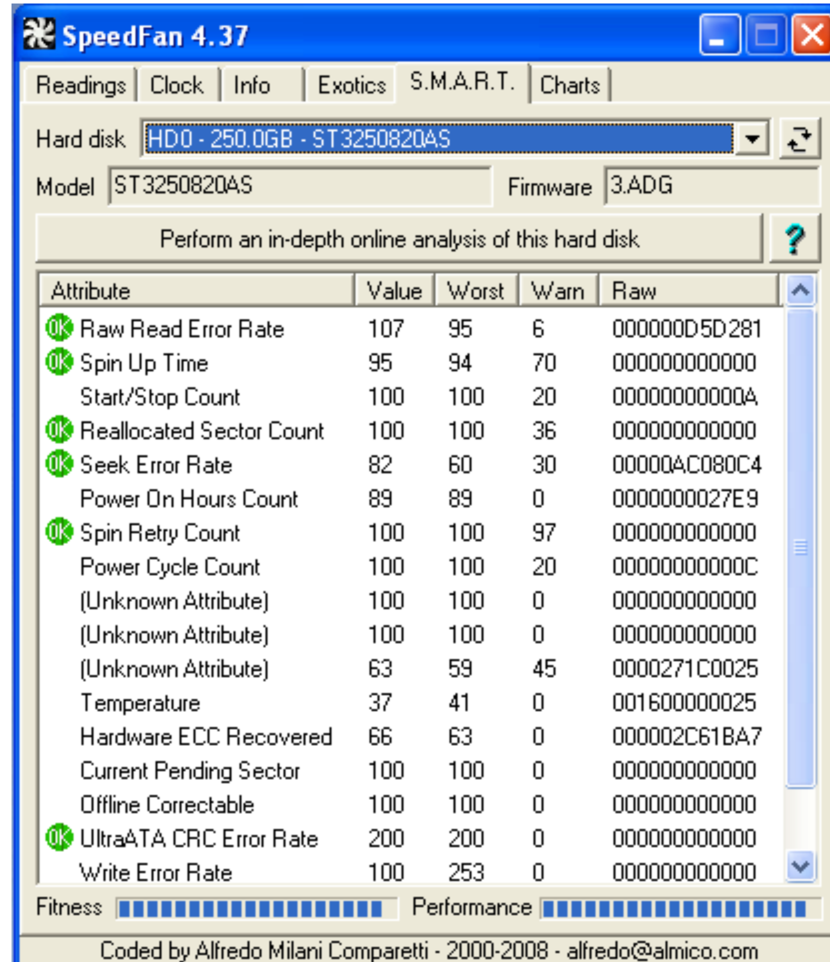
Nonsense values: lm-sensors is not properly configured on these machines.

Sensor monitoring for desktop: SpeedFan





	Attribute	Current	Raw	Overall
	Raw Read Error Rate	107	14013057	Good
	Spin Up Time	95	0	Good
	Start/Stop Count	100	10	Very good
	Reallocated Sector Count	100	0	Very good
	Seek Error Rate	82	180378250	Good
	Power On Hours Count	89	10216	Watch
Warning: *Power On Hours Count* is below the average limits (98-100).				
	Spin Retry Count	100	0	Very good
	Power Cycle Count	100	12	Very good
	Unknown attribute 187	100	0	Very good
	High Fly Writes	100	0	Very good
	Airflow Temperature	63	656146469	Good
	Hardware ECC Recovered	67	43916227	Good
	Current Pending Sector	100	0	Very good
	Offline Uncorrectable Sector Count	100	0	Very good
	Ultra DMA CRC Error Rate	200	0	Very good
	Write Error Rate	100	0	Very good
	TA Increase Count	100	0	Very good

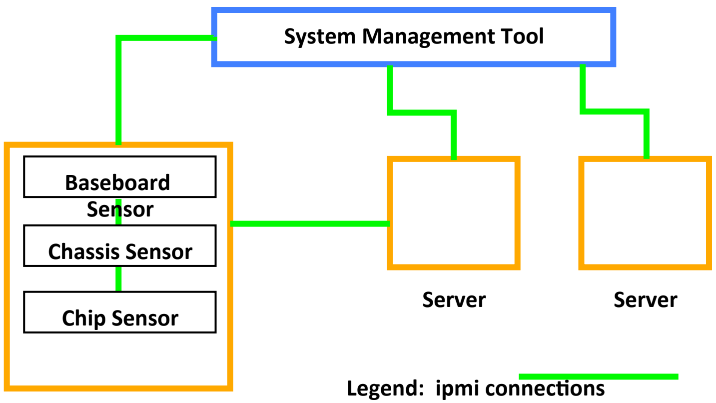


Attribute	Value	Worst	Warn	Raw
Raw Read Error Rate	107	95	6	000000D5D281
Spin Up Time	95	94	70	000000000000
Start/Stop Count	100	100	20	00000000000A
Reallocated Sector Count	100	100	36	000000000000
Seek Error Rate	82	60	30	00000AC080C4
Power On Hours Count	89	89	0	0000000027E9
Spin Retry Count	100	100	97	000000000000
Power Cycle Count	100	100	20	00000000000C
(Unknown Attribute)	100	100	0	000000000000
(Unknown Attribute)	100	100	0	000000000000
(Unknown Attribute)	63	59	45	0000271C0025
Temperature	37	41	0	001600000025
Hardware ECC Recovered	66	63	0	000002C61BA7
Current Pending Sector	100	100	0	000000000000
Offline Correctable	100	100	0	000000000000
UltraATA CRC Error Rate	200	200	0	000000000000
Write Error Rate	100	253	0	000000000000

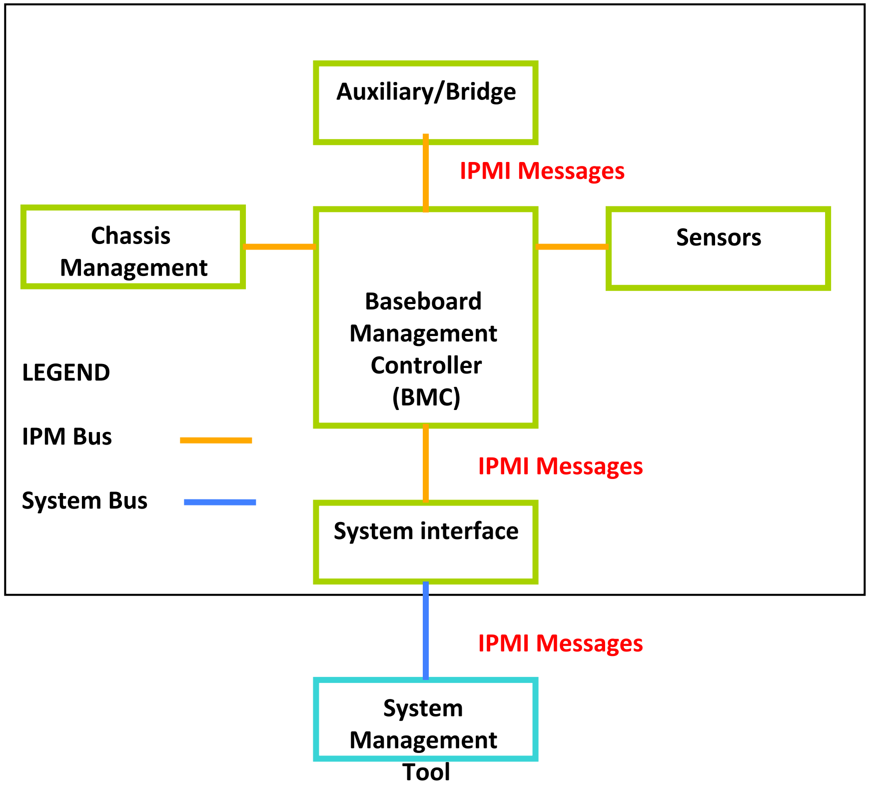
How would you do this with 500K cores ? You don’t.

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

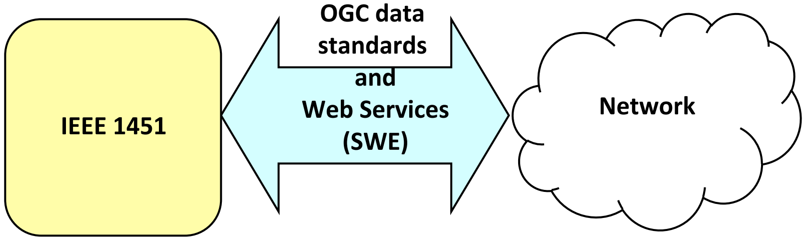
- **The Specification is Open Source (latest v.2.0)**
- **Tool implementation takes place at the BIOS level**
- **Requires proprietary drivers**

IEEE 1451-0: Standard for a Smart Transducer Interface for Sensors and Actuators

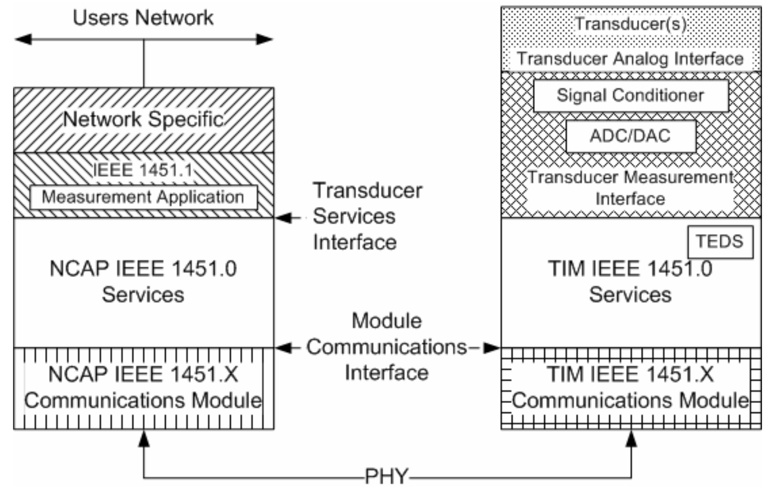
What is a "sensor" according to IEEE 1451?

- **transducer**: A device that converts energy from one domain into another. The device may be either a sensor or an actuator.
- **actuator**: A transducer that accepts a data sample or samples and converts them into an action.
- **sensor**: A transducer that converts a physical, biological, or chemical parameter into an electrical signal.

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

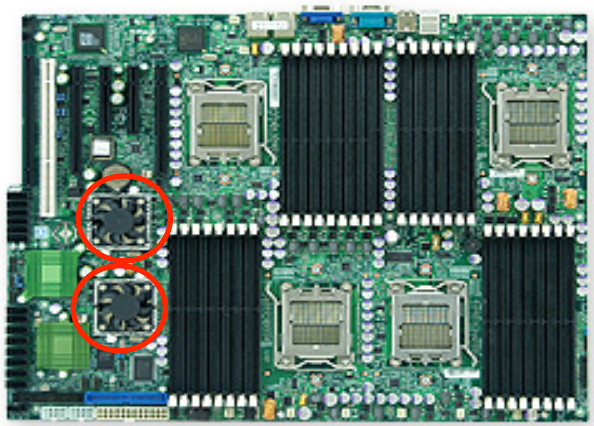
NCAP: Network Capable Application Processor

PHY: Physical Connections

ADC: Analog-to-Digital Conversion

DAC: Digital-to-Analog Conversion

Location counts



(POD1) Supermicro H8QM3

(PID1) Supermicro X7DWA

Intel D946GZIS

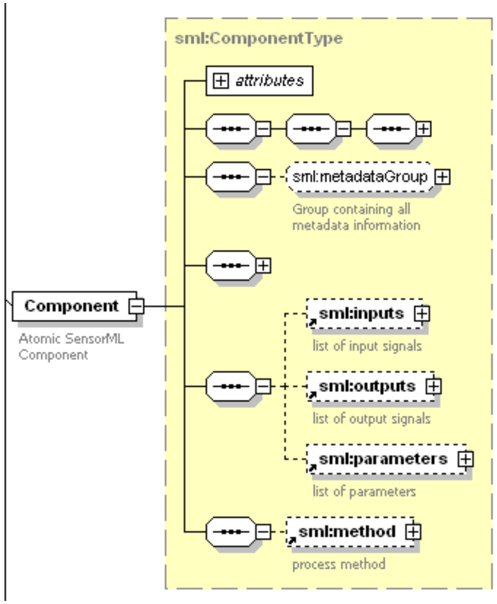
- Location of Field Replace-able Unit (FRUs)
  - GPS coordinates, RFID tags, location of a blade, etc.
  - all need to be represented

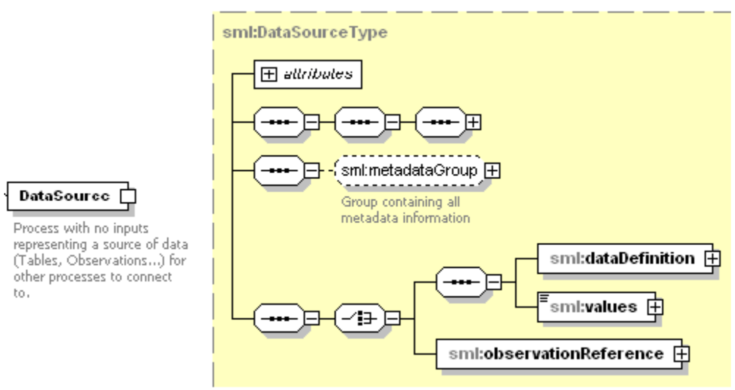
Open Geo-spatial Consortium: Sensor Model Language (Sensor ML)

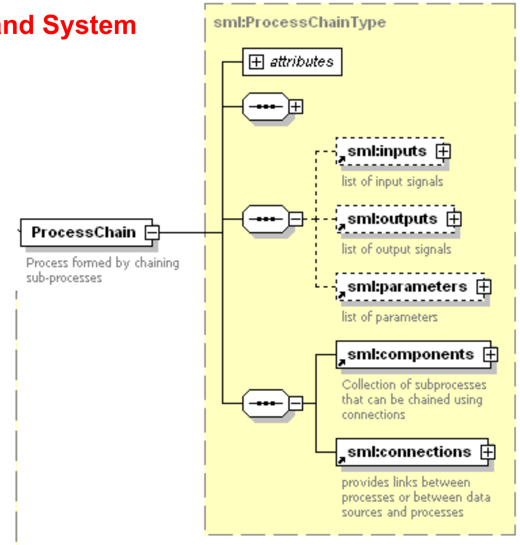
- One of a suite of standards providing methods for discovering sensors, accessing data, tasking sensors, receiving alerts over a network.
- A mark-up language for describing sensor systems, processing algorithms, workflows, location, and remote control of sensors.
- All entities are described in terms of processes:
  - A sensor is a process that converts a real phenomena into data.
  - Detectors, actuators, data readers are atomic physical processes
  - Spatial transforms, derivable information are non-physical, atomic processes
  - Provenance (history) is a non-physical, composite process
  - Sensor systems and platforms are physical, composite processes

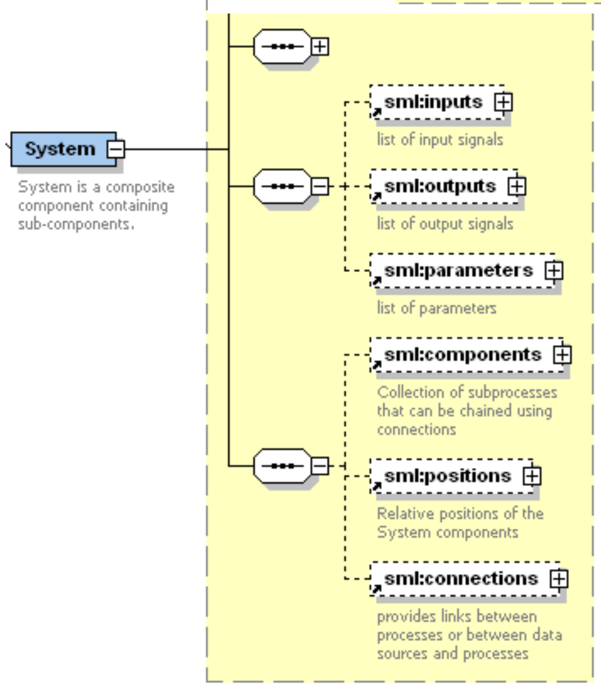
Sensor Types in Sensor ML

Component, Component Array, Data Source, Process Chain, Process Model, and System









Data structures in SensorML (e.g.)

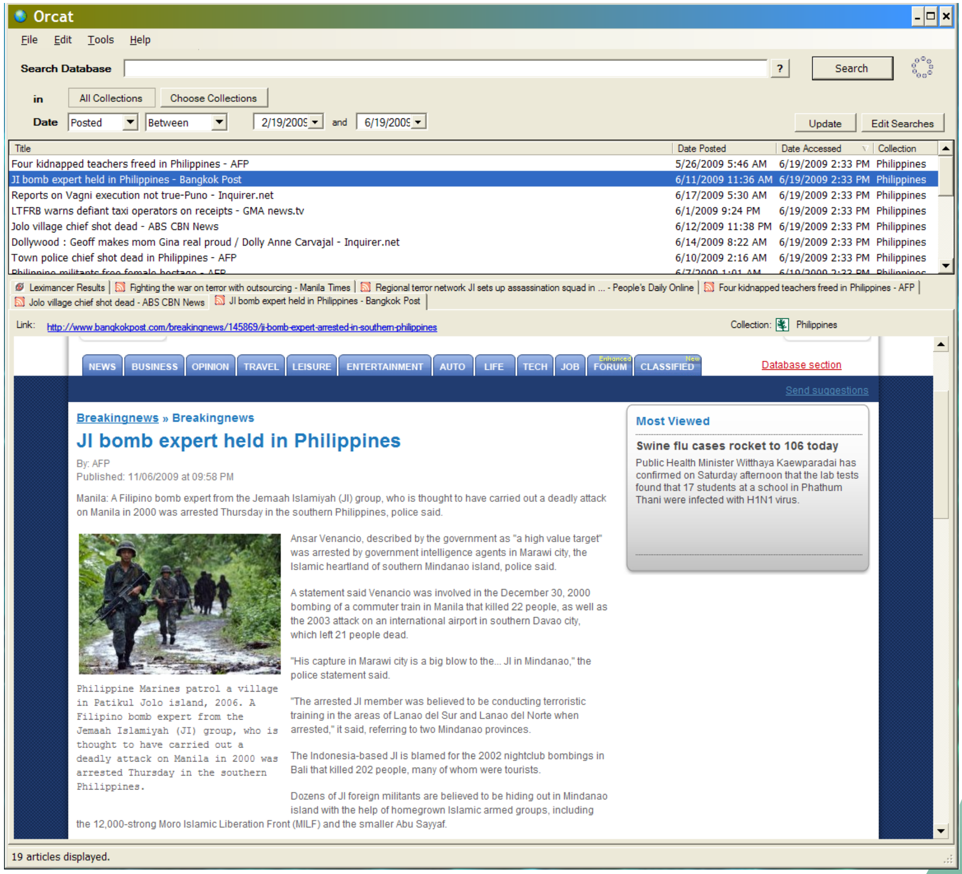
Elements for input and output signals, and parameters	
Name	**Description**
swe:Count	Integer for counting a value
swe:Quantity	Decimal with optional unit and constraints
swe:Time	Either ISO 8601 or time relative to an origin.
swe:Boolean	True or False
swe:Category	Identifies the name of a category, should provide dictionary entry for a useful interpretation.
swe:Text	Free text
swe:QuantityRange	Decimal pair for specifying a quantity range with constraints.
swe:CountRange	Integer pair for specifying a quantity range with constraints.
swe:TimeRange	Time value pair specifying a time range.
swe:ConditionalData	List of conditional values for a property.
swe:Conditional Value	Qualifies data with one or more conditions.
swe:DataRecord	Implementation of ISO-11404 Record Data Type.
swe:Envelope	Typically used to define rectangular bounding boxes in any coordinate system.
swe:GeoLocationArea	Area used to define bounding boxes.
swe:Position	Position given as a group of vectors/matrices.
swe:SimpleDataRecord	Implementation of ISO-11404 Record Data type.
swe:Vector	The Vector has a reference frame in which the coordinates are expressed.

Outline

- Data Collection at ORNL

How DCAT started: ORCAT

- A collection tool for Open Source Intelligence.
- Based on RSS feeds.
- Data stored in a database.
- Collected Web data and climate data (land-based observations).



System Sensors

- We are able to study the Trident cluster as a physical object
  - 32 nodes with quad-core AMD Opteron processors, a GigE Infiniband switch and a Myrinet switch.
- Only a few users currently running jobs
- Each node is essentially identical and equipped with the same set of sensors
- We are currently obtaining data from all the sensors we are aware of

The Sensors

- **Each of the 32 Trident compute nodes is equipped with at least 12 sensors**
  - **6 fan speed sensors (only 2 appear to give distinct readings)**
  - **6 temperature sensors (Recently obtained data from HP regarding location of sensors)**

Details from HP:
#1: near heatsink
#2,#3: next to each socket
#4: under power supply cage
#5: near power supply
#6: edge of board

IPMI – Intelligent Platform Management Interface Specification

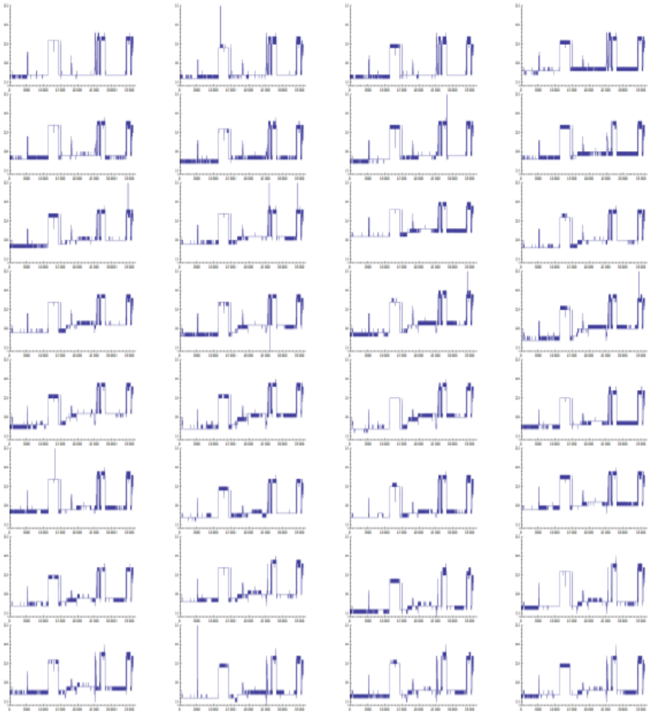
- Specification initially created by Intel
- Promoted by Dell, HP, Intel, NEC, adopted by more than 200 OEMS (not Cray)
- Open source specification, requires drivers and proprietary implementations
- Sensor data is queried via *ipmitool* – cron job outputs data once per minute
- Currently have ~25 million data points

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
- In contact with HP to determine source of errors

Current Sensor Data Set

- We have generally been running a single job at a time across the entire cluster
- So we generally expect very similar physical behavior from the nodes

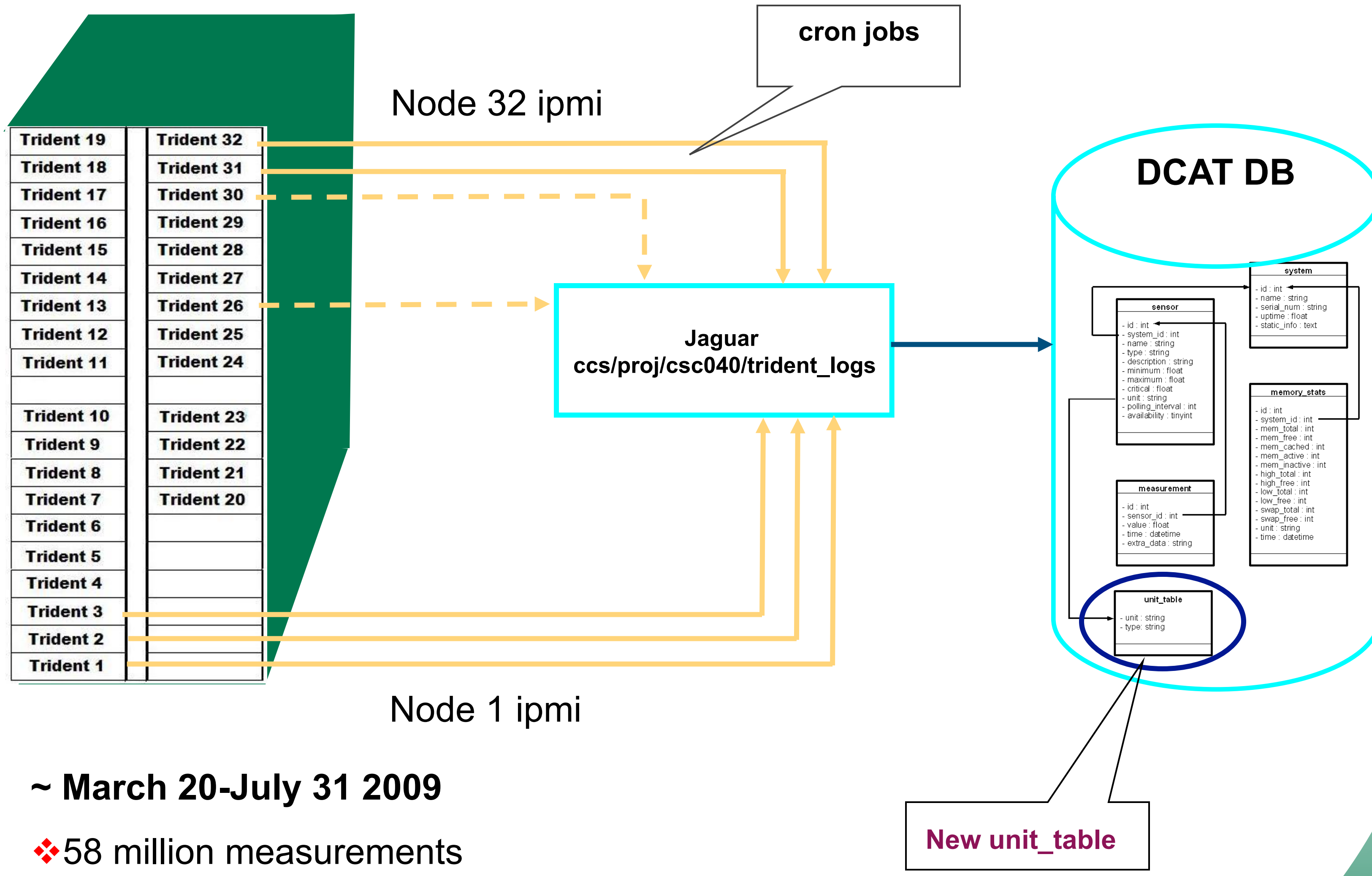


Heatsink Temperature Sensor
Trident 1-32

Examining the Sensor Data

- The "equilibrium" temperature is now about 3 degrees higher now for each sensor - is the data center warmer?
- Trident11 is consistently 3-4 degrees (C) warmer than the others
  - On top of the IB switch
- Trident24 sits in the same position in the other rack
  - Not as hot, but power supply failed

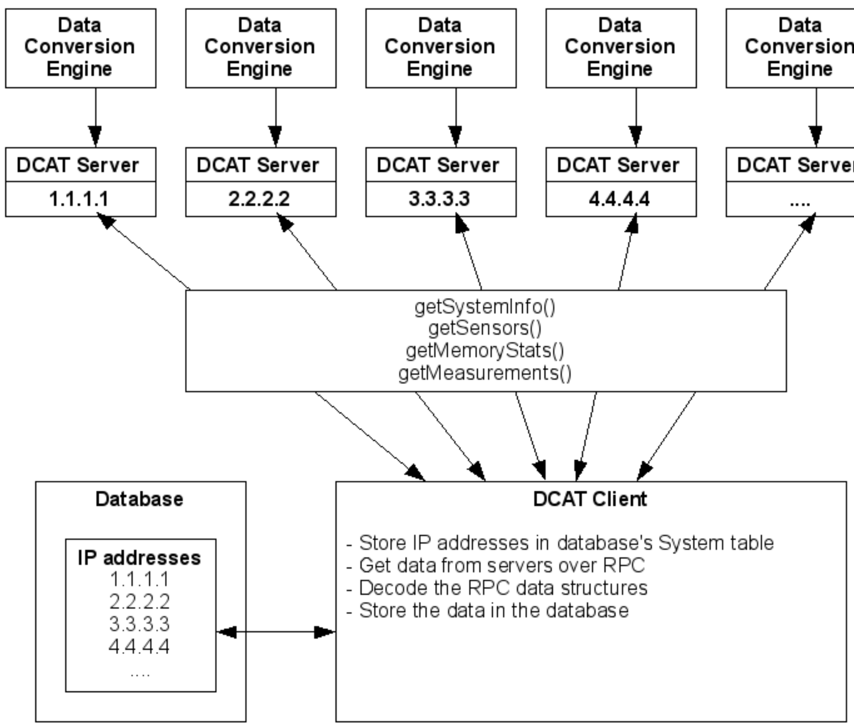
Trident nodes sensor data

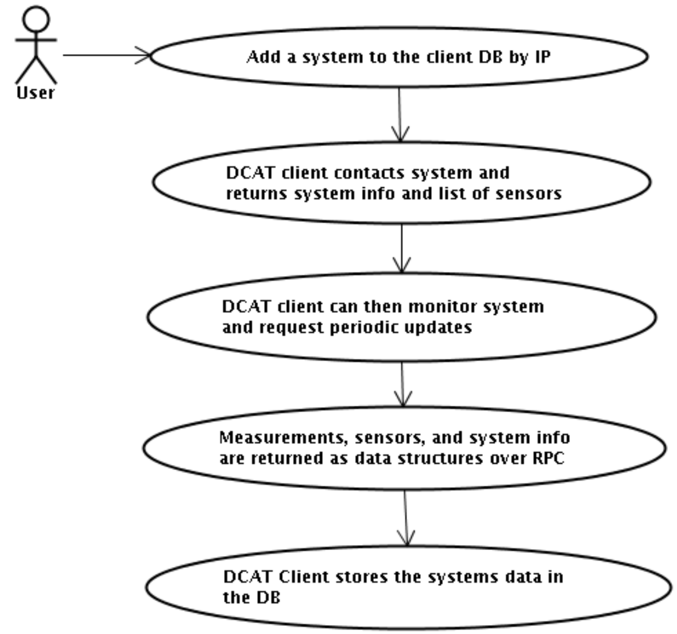


~ March 20-July 31 2009

- 58 million measurements
- 10GB

DCAT client-server architecture

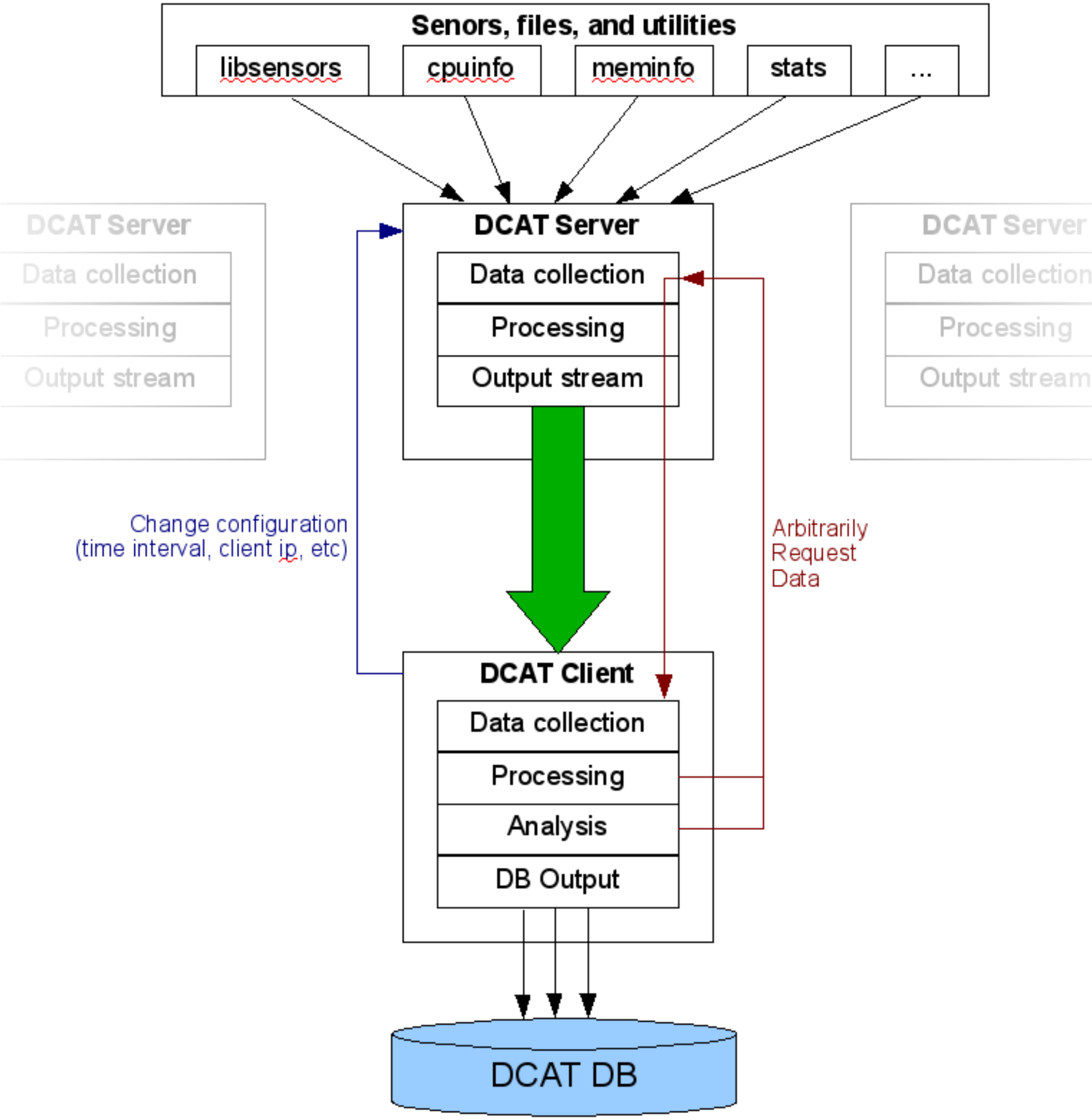




- Multiple sensors at each node
- One instance of DCAT server at each node
- Performs sensor data conversion

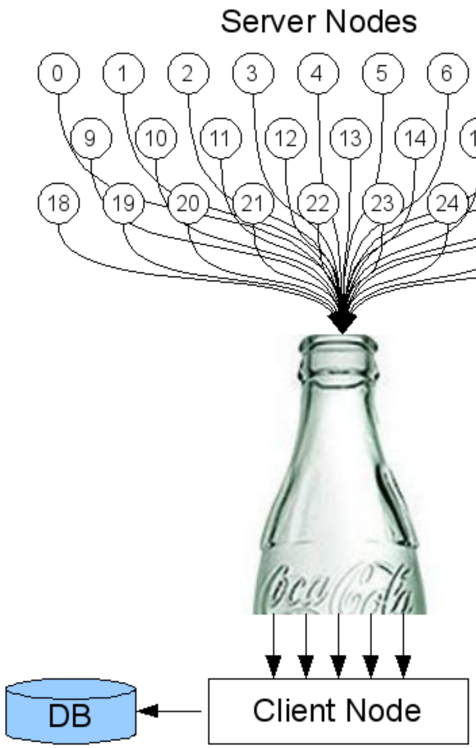
DCAT Architecture

- 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.



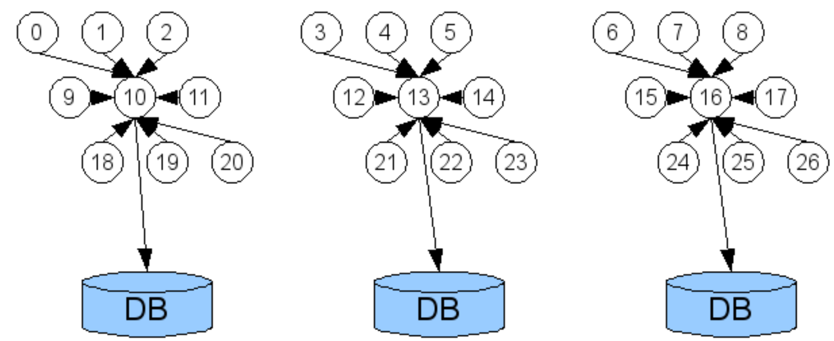
Scalability

- With only one client node, a bottleneck occurs.
- Solution:
  - Split nodes up into small groups.
  - Use multiple client nodes.
  - Use multiple databases.



As the number of nodes increases, DCAT will experience a bottleneck with all the server nodes trying to push data to a single client at once.

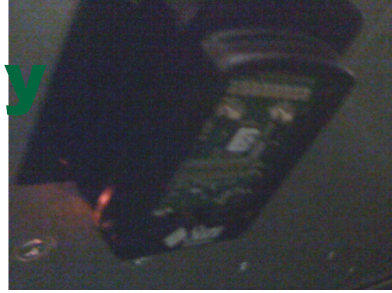
To remedy this, we could have multiple client nodes, with a small number of servers 'attached' to each client. Each client may then write data to either a single global distributed database, or their own local database. (as shown below)

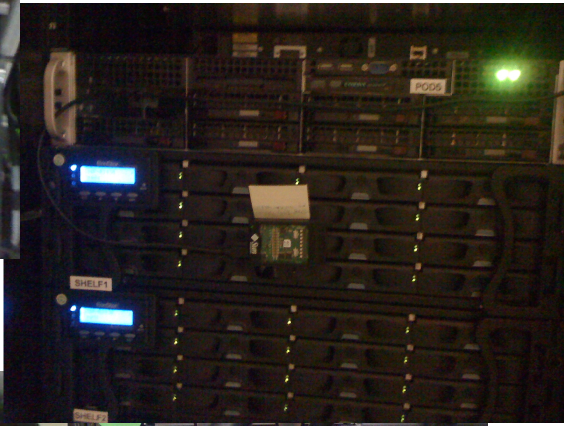


Outline

- **Validation: external sensors**

Sun™ Programmable Object Technology (SunSPOTs)

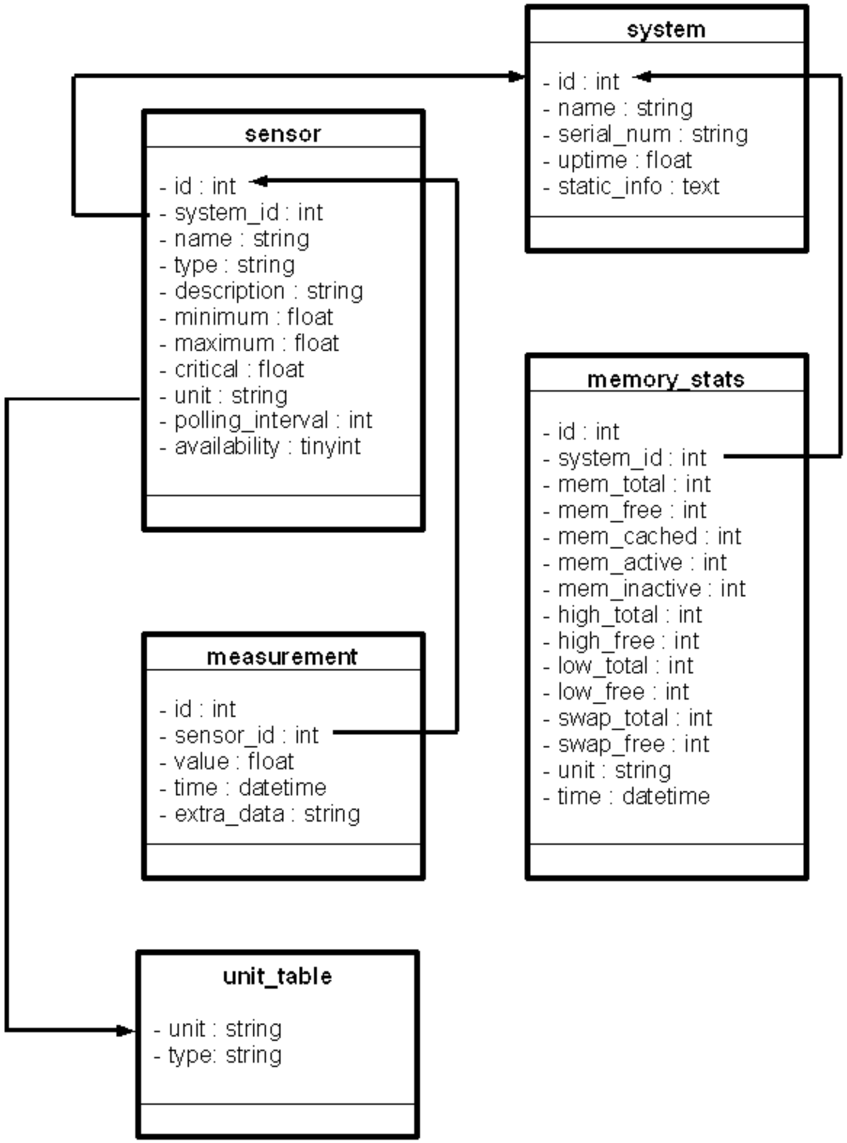




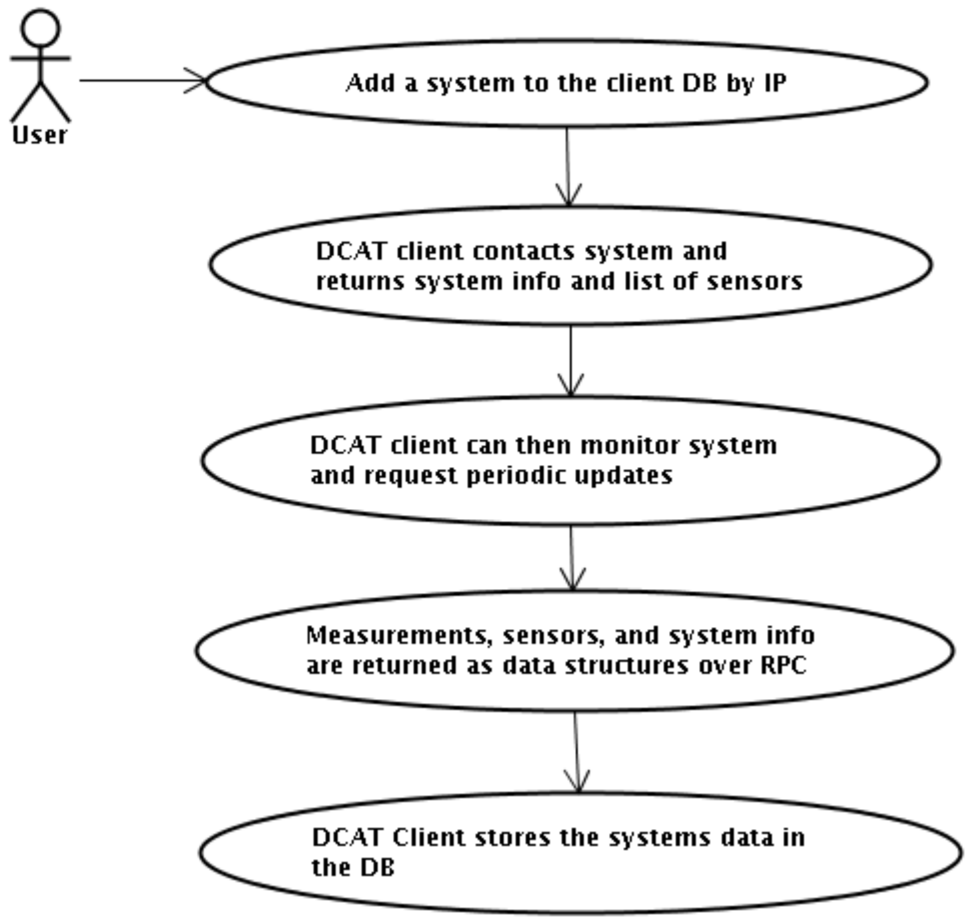
- 41x23x70 mm, 54gr., 40ft of range.
- Powered by USB.
- Contains an Accelerometer, a light sensor, and a temperature reader.
- Communicates with a base-station (same size just a little thinner) with its own IEEE 802.15.4 wireless network.
- Remotely programmable through the base-station, itself a USB device on Trident Management Node 2.
- Uses JRE and Ant technology (java based).
- Installed on Trident InfiniBand switch, Infotrend, Altix ICE, CB 11 RAID, IBM System P5.

DCAT data model

- **New DCAT database table: unit_table**

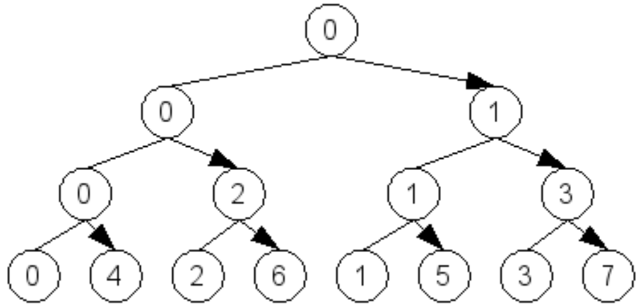


Client flow chart



Using MPI for Communication

MPI_Bcast



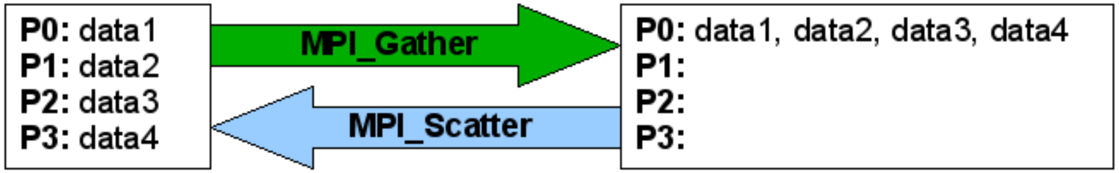
Useful for distributing changes to the server nodes or sending a request for data to the server nodes. Allows us to distribute data in $\log_2(p)$ stages instead of p-1 stages.

When p=1024, that means 100 stages rather than 1023 stages.

MPI_Reduce

MPI_Reduce basically does the reverse of MPI_Bcast, but performs a given operation (min, max, add, mult, etc..) on the data, leaving the receiving node with the final answer. This will be most useful when doing analysis on the fly. For instance, getting the average system load or solving an optimization problem.

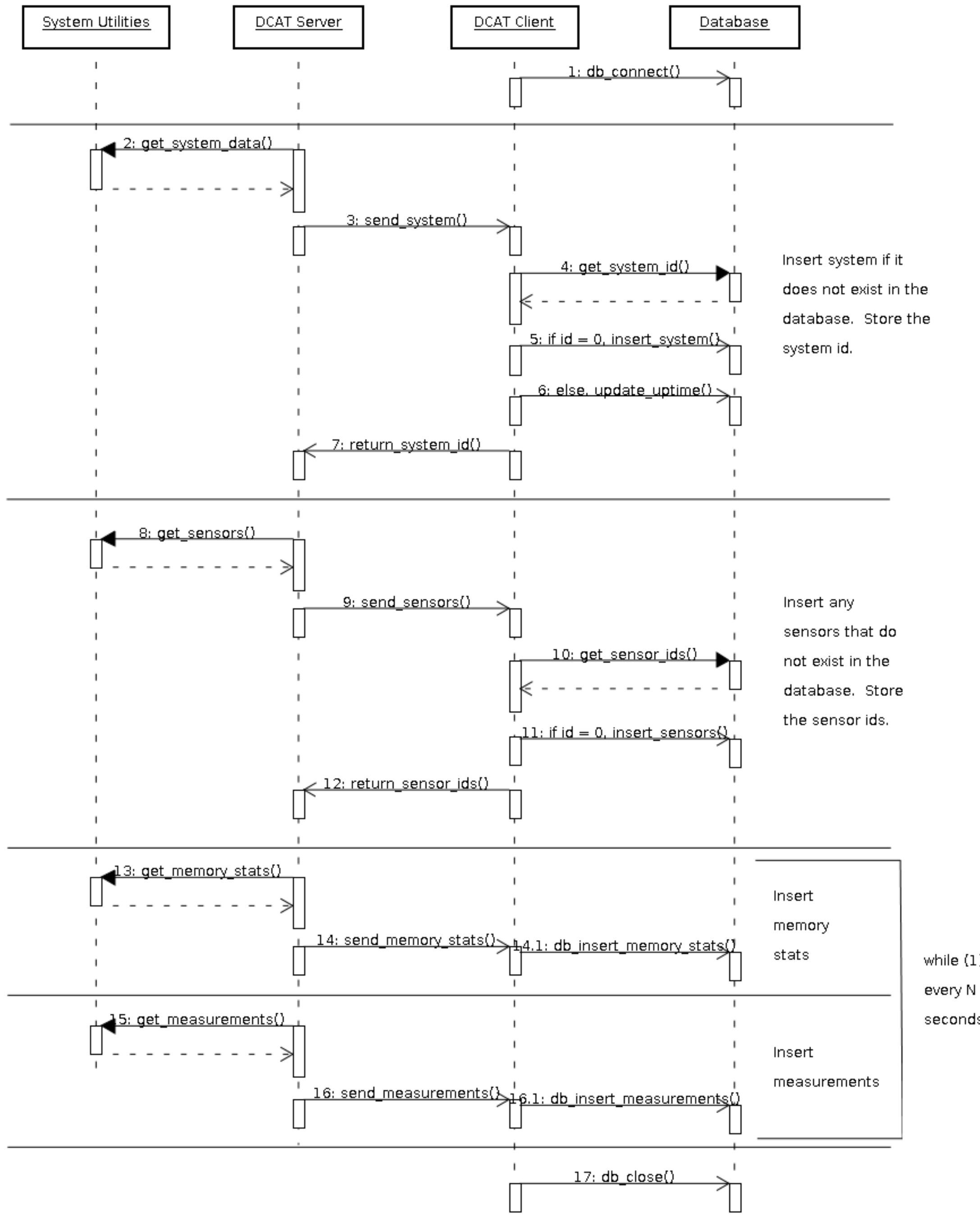
MPI_Gather and MPI_Scatter



MPI_Gather is the easiest way to get data sets from N processes to a single process. This can be used to collect data from the server nodes in order to add it to the database. MPI_Scatter does the opposite, distributing N data sets to N processes.

MPI Sequence Diagram

- Diagram shows the communication in detail.
- All functions shown are implemented.
- Currently working to clean up and optimize the code.



Outline

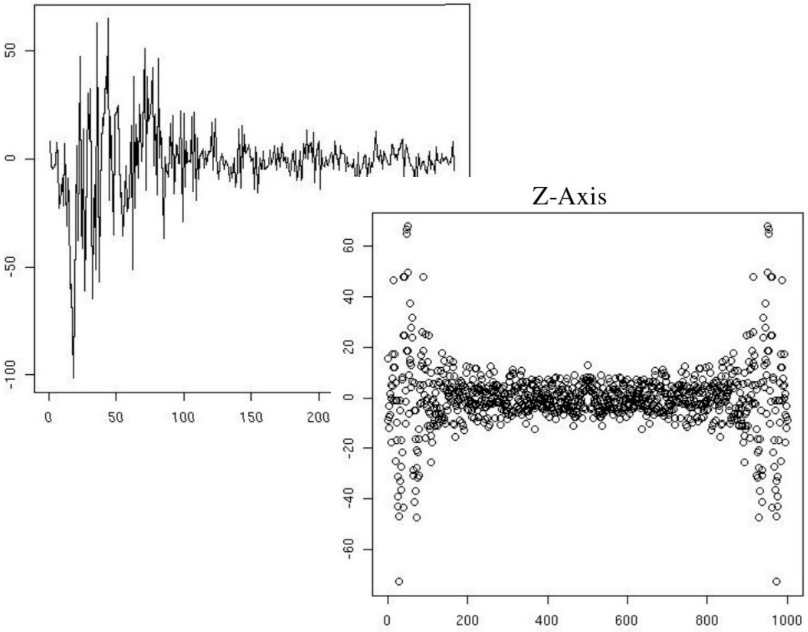
- **Visualization**

Evaluation of visualization environments

- **Cacti, based on Round Robin Tools:**
  - Plots linear time series.
  - Advanced visualization with aggregated data cannot be represented.
- **GNUplot utility does not enable interaction with data.**
- **I-seismometer data** (courtesy PJ Kennelly, Long Island University)

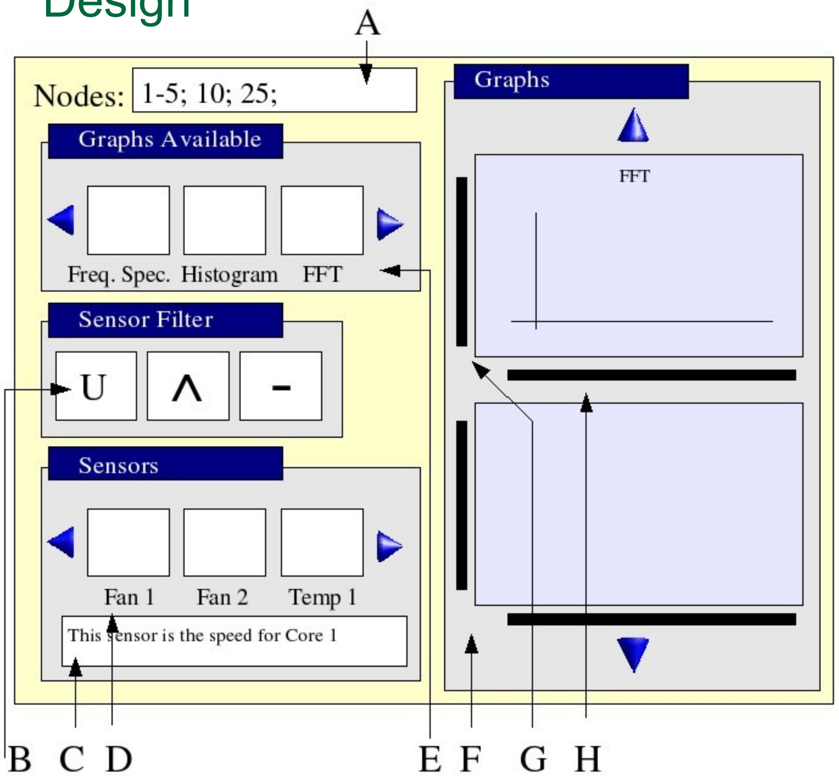
Histogram of x, y, z axes on i-seismometer.

Spectrum: intensity over frequency



Visualization for DCAT

Design



A: Enter target node (s).

B: Query operators for sensor selection.

C: Sensor description.

D: Sensor name.

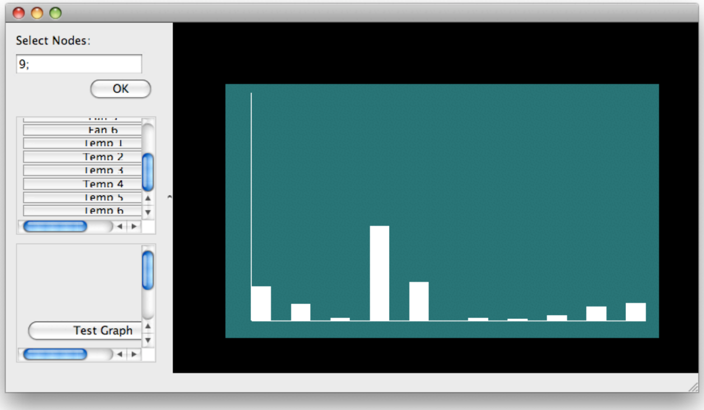
E: Available graphs.

F: Graph views.

G: Y scaling.

H: X scaling.

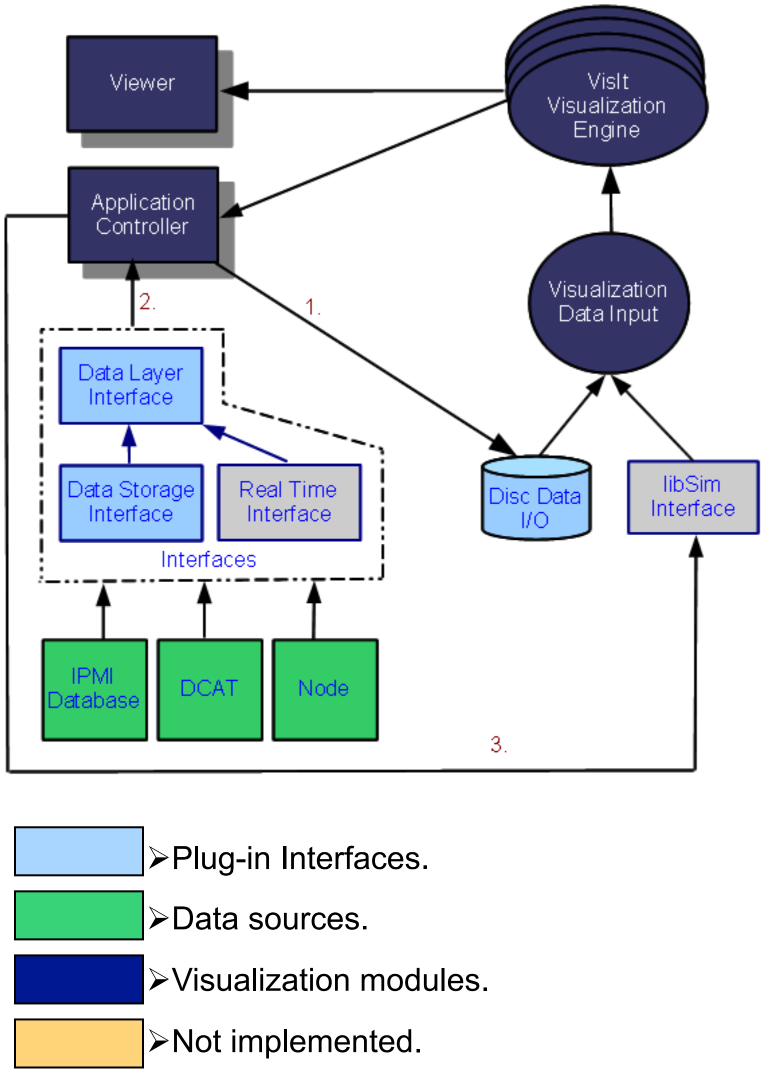
Preliminary results



Histogram for node 9, temperature sensor 6, 10 buckets (42-51 C), 59872 temperature readings, (March 20-May 2nd).

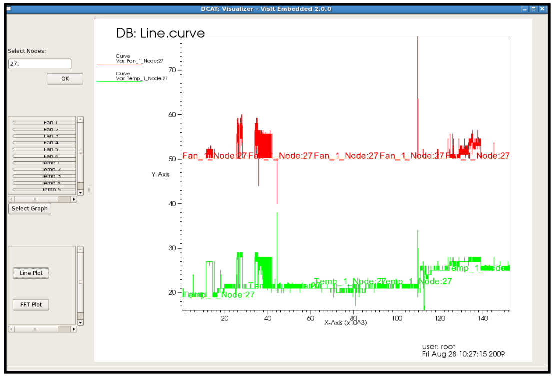
VisIt

- Developed by the Department of Energy (DOE) Advanced Simulation and Computing Initiative (ASCI) to visualize and analyze the results of terascale simulations.
- An analysis tool that provides the capability to calculate derived fields.
- Decoupled architecture including Viewer, Engine, and plug-ins.
- Open Source.

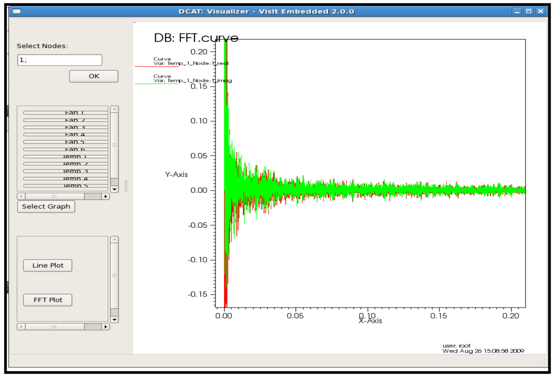


DCAT data displayed with the VisIt Plug-in

- **The DCAT plug-in uses the VisIt Engine.**
- **The Plug-in enables an API to the DCAT database.**
- **Includes a User Interface.**



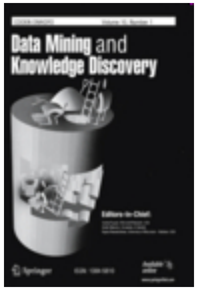
Temperature and Fan data for Node 1



Temperature Frequency Data

Knowledge Discovery

- A body of research focused on extracting information by recognizing patterns from very large datasets and transforming it into actionable knowledge by presenting it, organizing it, and integrating it into its context.
- Theoretical foundations, methods, algorithms, best practices, standards, specifications.
- Journals, SIG groups, conferences, books.
- Applications in science, medicine, intelligence, and many other domains.
- Interdisciplinary research originated in data mining.



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.