Monitoring Cray Cooling Systems

Don Maxwell, Matthew Ezell, Matthew Donovan and Christopher Layton
National Center for Computation Sciences
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
{mii, ezy, md5, clp}@ornl.gov

Dr. Jeffrey Becklehimer
Cray Inc.
Oak Ridge, Tennessee, USA
jlbeck@cray.com

***Abstract*—While sites generally have systems in place to monitor the health of Cray computers themselves, often the cooling systems are ignored until a computer failure requires investigation into the source of the failure. The Liebert XDP units used to cool the Cray XE/XK models as well as the Cray proprietary cooling system used for the Cray XC30 models provide data useful for health monitoring. Unfortunately, this valuable information is often available only to custom solutions not accessible by a center-wide monitoring system or is simply ignored entirely. In this paper, methods and tools used to harvest the monitoring data available are discussed, and the implementation needed to integrate the data into a center-wide monitoring system at the Oak Ridge National Laboratory is provided.**

I. Introduction

Monitoring the Cray cooling systems at the Oak Ridge National Laboratory (ORNL) has primarily been the responsibility of Cray hardware engineers relying on isolated systems that generally only use email for notification. While this might be sufficient for some sites, a somewhat complex monitoring system has been developed at ORNL with human monitoring available on a 24x7 basis. With no insight into the Liebert XDP units cooling most of our Cray systems in the central monitoring system, a decision was made to attempt to bridge this gap.

The Liebert XDP units natively speak a data communication protocol for Building Automation and Control Networks called BACnet. This is a global standard used in the refrigeration, air-conditioning and heating industries. The obvious problem is that most open source monitoring solutions do not speak BACnet. Many of them natively speak SNMP which is the primary protocol used at ORNL for monitoring, so solving the XDP monitoring problem became a matter of simply finding a way to translate BACnet into SNMP. Two solutions were explored to attempt to resolve this problem, and experiences with both will be discussed.

Tying the Cray proprietary cooling system used for the XC30 into the central monitoring system can be accomplished in several ways. The monitoring data can be obtained using the existing System Environment Data Collections (SEDC) server, or it can be obtained directly via command line. The focus to this point has been dedicated to solving the Liebert XDP monitoring since those units provide the cooling for the premier Cray XK7 Titan system at ORNL, but work is underway to accomplish the same goal for the Cray XC30 Eos system at ORNL.

Once the monitoring data can be harvested, it must be presented to the monitoring system in order to be useful. As mentioned earlier, the primary protocol used at ORNL to communicate to the monitoring system is SNMP. Using SNMP, many custom scripts and queries have been developed to allow the central Nagios monitoring server to communicate with the Cray hosts to provide system status. Similarly, using custom SNMP scripts and queries, the cooling system data is presented to the Nagios server to provide status, alarms outside of thresholds, etc.

Finally, with access to this data, graphs can be generated to look for outliers and trends. While specific thresholds, alarms, and status can be communicated using Nagios, problems such as hot spots, high humidity, high chilled water temps in a portion of a computer room, an underperforming cooling unit, etc. can most easily be found by seeing all of the data in one graph. Using the Multi Router Traffic Grapher (MRTG) [1], each data point is charted for individual units as well as all units. This provides a quick glance for potential problems as well as trends for values that could point out developing problems. Example charts will be provided to demonstrate the value of data presentation in this form.

II. Motivation

Recent Cray systems arriving at ORNL have been installed with a custom cooling solution not provided by the site. Historically, most ORNL systems were cooled by computer room air conditioning (CRAC) units provided by the site. These units blew huge volumes of air underneath a raised floor with holes cut in the floor to provide the air to the system for cooling. This method of cooling is obviously

very inefficient. The cool air cannot be easily directed to the intended target because the area under the raised floor is typically very open and imprecise holes in the floor allow more air to escape than needed in certain areas. Attempting to compensate for these inefficiencies can also create a very cold room.

More efficient cooling systems have now been developed for most large systems that in some way or another connect the site's chilled water system more directly to the system. Typically, this employs the same methods used by CRAC units whereby chilled water is used to cool a refrigerant with fans then blowing across coils containing the refrigerant to provide cold air. The difference is that the refrigerant and fans are contained within the system itself to localize the cold air rather than having air blown indirectly at the system providing a much less efficient delivery method.

The Liebert XDP units used to provide cooling for the Cray XE/XK models accommodate this localized cooling method and also contain sensors to provide monitoring data for the units. Based upon the particular Liebert model, a couple of methods are available for retrieving this data. First, for older model Liebert XDP units such as the ones onsite at ORNL, serial interfaces from each XDP can be consolidated into a device called a SiteLink-E, which provides a 10/100 Mbps Ethernet port for external communication. The particular SiteLink-E model at ORNL has twelve serial ports that allow twelve XDPs to be presented via a single Ethernet interface. Second, an interface card called the Liebert Intellislot Web Card can be purchased for later model XDPs to provide a network interface for each individual XDP. The Intellislot allows access to the monitoring data using the SNMP protocol. Liebert also provides a software package called SiteScan that allows users to monitor and control the Liebert XDP units. This package is fairly comprehensive providing monitoring, alerts, reports, graphs, etc.

As mentioned above, ORNL uses a Liebert XDP model that does not support the Intellislot cards. While the SiteLink-E devices provide access to the XDPs via Ethernet, they only support the BACnet protocol – not SNMP. Cray purchased the SiteScan software package that speaks BACnet to provide monitoring, alerts and graphs. While this may work well for some centers, it doesn't necessarily fit well in an established ecosystem for several reasons. SiteScan runs on an isolated PC running Windows. The Oak Ridge Leadership Computing Facility (OLCF) network does not maintain an infrastructure for support of operating systems other than Unix, so the SiteScan PC could not be attached to the network providing existing monitoring infrastructure. This obviously creates several integration problems including the fact that this would introduce another software package into an existing monitoring infrastructure. While SiteScan does provide the ability to send SNMP traps on alerts, notifications via email, graphing and reporting, and other features, this becomes suboptimal when the machine is unreachable from the network and even overcoming that, again introduces a new monitoring system into an existing infrastructure.

III. Liebert XDP Monitoring for Titan

A. *Production Monitoring using Commercial Device*

When exploration of ways to bridge the BACnet/SNMP gap discussed earlier began, a commercial device was identified which seemed to fulfill all of the requirements. The QuickServer Building Automation System Gateway from FieldServer Technologies [2] provides the necessary protocol translation to allow SNMP queries for BACnet data points on the Liebert XDP units. Attaching this device to the same Ethernet network that connects all of the SiteLink-E devices provides the necessary communication to make the translation as illustrated in Figure 1. Using a QuickServer configuration that identifies each SiteLink-E device via IP address, any single data point on a particular XDP can be queried or managed. Due to potential security issues with the QuickServer and SiteLink-E devices, the QuickServer at OLCF resides on a private network with a Cray System Management Workstation (SMW) providing the link to the outside world. This was simply a convenient machine for making the connection and has no particular software required that couldn't exist on any machine.

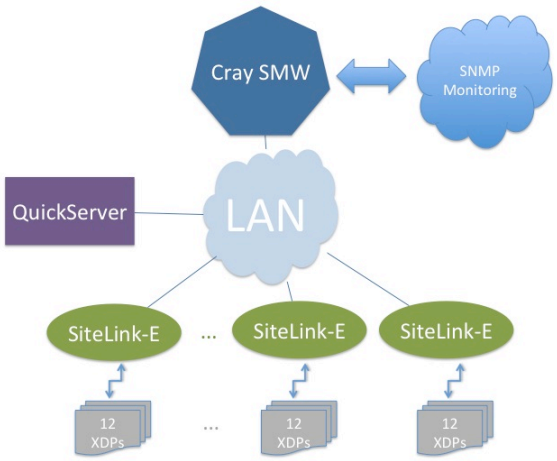

Figure 1. Liebert XDP Network

While this configuration has worked very well once established, there were some issues in getting to production. First, there is a limit to the number of data points that can be specified for the QuickServer. There are probably some hardware limitations that would eventually be encountered as the number of data points increase, but marketing is probably the main driver behind limiting the number of data points since the cost increases based on the number of data points configured. This also seems to be the driver behind the requirement of providing the data points to FieldServer

in order to obtain a configuration for the QuickServer. Each configuration is a custom configuration provided by FieldServer that defines clients (i.e., SiteLink-E devices for OLCF), and a mapping of each data point to object ids for twelve XDP units being served by a single client. A SNMP MIB is also provided with the configuration. The obvious disadvantages to this requirement are the slowdown in getting to production since support must be involved to begin working with the device, the inability to make changes to data points based on different needs, and the inability to add new clients if needed. In general, the inability for the customer to make configuration changes without involving technical support is disadvantageous. If the device is being used in a somewhat static environment, as is the case with the XDPs, it is probably not a huge issue.

OLCF chose the following BACnet data points to be monitored or set by the QuickServer.

- bs01_valve_percent_open
- bs08_local_temp
- bs09_local_humidity
- bs10_local_dewpoint
- bs15_high_chilled_water_temp_setpoint
- bs04_fluid_temp
- bs05_chilled_water_temp
- g406_valve_failure
- g409_high_chilled_water_temp
- g418_low_pressure
- g419_high_refrigerant_temp

In consultation with the Cray hardware staff, these eleven data points were identified as the most critical for indicating existing or potential future problems. Based on the descriptions, it is easy to see the reasoning for selection of these points. At times, it is necessary to use two or more points for problem determination since one alone does not necessarily indicate a problem but simply the system reacting to circumstances. A valve opening to 100% with no corresponding rise in the refrigerant temperature probably only means that a very efficient code is running and the system is reacting to keep the machine cool. Circumstances such as these need to be identified in order to configure a credible monitoring configuration.

Once the data points were defined and the configuration was obtained and loaded onto the QuickServer, a method for querying the data points via the SMW had to be designed. SNMP proxies have been used extensively at OLCF to monitor many Cray nodes and devices that have no external interface, so proxies were the obvious choice. Using SNMP proxies, a node or machine with an external connection can be used to provide SNMP data for another machine that shares a common interface. Referring to Figure 1 again, the SNMP monitoring network is able to query the QuickServer by using a SNMP proxy that has been established on the Cray SMW. OLCF uses the Nagios® [3] monitoring package to provide status of machines and services on a 24x7 basis. The Nagios monitoring server is able to directly query each data point for each XDP and alert operators and support staff if any problems are detected. This capability clearly provides the potential to identify and rectify problems with the XDPs before they become problems for the Cray itself.

B. Alternative Monitoring Using Open Source Package

While the vendor provided the QuickServer device to solve the XDP monitoring needs of the site, another solution does exist and seems to provide more flexibility than the QuickServer. An open source package called BACpypes [4] provides a python library that can be used to build BACnet applications. Using this package, the BACnet protocol translation can be accomplished to retrieve or set data points for the XDPs by communicating directly with each SiteLink-E. Given this part of the monitoring process is solved, it's then simply a matter of retrieving data points via scripts that can easily be called using SNMP. The BACpypes package provides sample python scripts that can easily be used to retrieve data points, so writing a new application to use the python libraries would not necessarily be required, but a different interface might be more desirable and could be accomplished with minimal effort. An example query using one of the sample scripts is included below in Figure 2.

```
smw$ ~/bacpypes/samples/ReadProperty.py
> help read
read <addr> <type> <inst> <prop> [ <indx> ]
> read 192.168.168.2 analogValue 10 description
> Fluid Temp
> read 192.168.168.2 analogValue 10 presentValue
> 53.2000007629
> read 192.168.168.2 analogValue 10 units
> degreesFahrenheit
> read 192.168.168.2 binaryValue 11 description
> High Refrigerant Temp
> read 192.168.168.2 binaryValue 11 presentValue
> inactive
> exit
Exiting...
```

Figure 2. Example ReadProperty.py Session

This script can also provide a simple way to query descriptions of data points to determine which might be of interest for monitoring. By walking both the analog values and binary values in a sequential fashion, each description can be queried along with associated attributes as described in the object.py source file. As could be assumed, analog values are typically data points that have a numeric value associated with them – temperature, percent open – while

binary values are generally true or false - set or not set as could be expected for alarms. By sequencing through the fourth argument representing the object identifier or instance, each description can be returned for potential inclusion in XDP monitoring.

In comparison, the BACpypes package is more flexible than the QuickServer device since it allows the user to change configurations without involving any support organization. The amount of work needed to use the package is minimal, and the corresponding scripts that would need to be written to harvest the data points is quite frankly the same as those needed to poll the QuickServer using SNMP. The package is quite robust, and if the package had been discovered before the vendor had supplied the QuickServer, OLCF would have probably used BACpypes in the deployment of the XDP monitoring software.

C. *Alerts for Nagios Monitoring*

Regardless of the data point delivery method, thresholds must be defined to provide triggers for Nagios alerts. While data points that provide alarms such as valve failure or low pressure require no thresholds, others such as local temperature or chilled water temperature need thresholds for notification. OLCF implemented a simple Perl script to allow specification of both warning and critical values or a range of values. Assuming there is only interest in one value, such as 100 percent for valve percent open, warning and critical values can be the same. A range can be specified for data points such as temperatures outside of which a Nagios critical alert is triggered. Again, in consultation with local Cray hardware staff, alert values were defined for each data point and Nagios configurations were then created which simply call the Perl script with appropriate arguments resulting in Nagios exit statuses based on the data point.

Additionally, the Perl script is able to provide a dynamic calculation of the SNMP Object Identifier (OID) due to the regular nature of the OIDs created for the data points on the QuickServer. This avoids the need to statically specify each OID when doing SNMP queries by simply using a few lines of code. Using the QuickServer, SNMP and Nagios, XDP monitoring is currently running in production at the OLCF.

IV. Cray XC30 cooling system monitoring

While Cray cooling system monitoring has been focused on Titan and the Liebert XDPs used for cooling it, some investigation into monitoring the cooling system for the four-cabinet Cray XC30 at OLCF has begun. The Cray XC30 has a proprietary cooling system with an entirely different design and interface. Traditional designs cool from top to bottom, but the Cray XC30 has a transverse cooling system that includes a blower assembly between pairs of cabinets. Each cabinet contains a chilled water loop to provide the cooling for the transverse air. Clearly, many of the same components exist in the cooling systems for both Cray models at OLCF, but the methods for retrieving sensor data are quite different. Since the XC30 cooling system is a Cray proprietary system, the tools needed to retrieve the sensor data are also Cray proprietary and reside on the SMW. Using a command called xtcheckhss, many of the same sensor data points that exist on the Liebert XDPs can be retrieved. In addition, xtcheckhss provides other data for the system itself including CPU temperatures, DIMM temperatures, Aries temperatures, voltages, air temperatures and velocities, humidity readings, etc. This wealth of data provides endless opportunities for monitoring the health of the system and correlation with jobs.

Following the same methods used to incorporate the XDP monitoring into the center, it should be fairly straightforward to add the XC30 cooling monitoring to the center-wide Nagios monitoring system. Again, in consultation with the Cray hardware staff, important data points needed for monitoring along with thresholds will be determined and afterwards, a method chosen to incorporate the output of xtcheckhss into SNMP for data retrieval by Nagios. Some thought will need to go into how this might be accomplished most efficiently. It appears that xtcheckhss provides the functionality to pinpoint data points given the right arguments, but more research is needed to determine if that is truly practical. If so, it could be as simple as running the command directly on SNMP poll. Otherwise, a complete dump of the output of xtcheckhss at intervals might be necessary with a mapping to SNMP OIDs for retrieval. There is ongoing research into the capabilities of the tools needed to retrieve the data points.

V. Graphs and Data

While all of this data provides a robust system for health monitoring, it can also be the source of discovery for other useful applications. The MRTG package provides a very useful visualization of data and manages the data keeping the storage needs constant over time by using RRDtool [5]. MRTG was developed to monitor and measure load on network links, but it can obviously be used to graph any data. It natively speaks SNMP, which provides another motivation for exposing all data points using SNMP. Often, it can be difficult to see trends in data without a visual picture. A gradual increase in data points for a particular component or particular area of the room will not necessarily trip an alarm but might indicate a potential problem that needs addressing. Trends can easily be seen when the data is graphed and grouped and even overlaid for correlation.

Since the XDP monitoring system was put into production in January 2014, there have only been a small number of alarms. First, one particular XDP near the center of the Titan XDP units typically falls 1 or 2 degrees out of local temperature range when the machine is shutdown. Clearly, this is no cause for concern, but possibly reflects a portion of the room that tends to get slightly colder than the rest of the room when Titan is not producing a heat load or

maybe a somewhat faulty temperature sensor. Looking at the trend of temperatures, an interesting observation is made. Figure 3 clearly indicates that this particular XDP runs at a lower temperature than the other XDPs under all circumstances. The fact that it dips below the threshold when there is no load is much less surprising after observing the data that has been collected since the XDP monitoring system was put into production. A graph of the data provides a quick conclusion at a glance. A couple of possible outcomes are either further investigation into the temperature sensor for that XDP, an environmental survey near that XDP, or simply adjusting the lower threshold for this XDP since it is only one or two degrees below threshold.

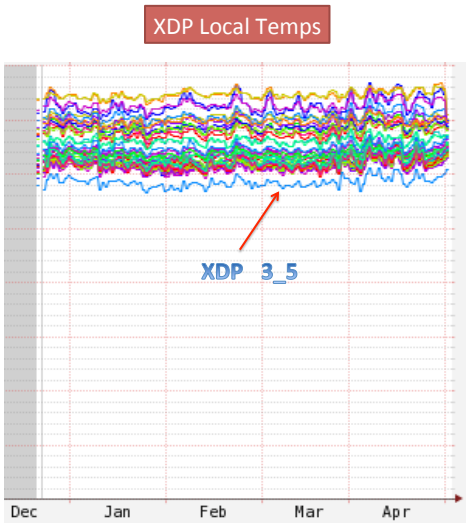

Figure 3. XDP Local Temperatures

Second, on three different occasions, particular XDP chilled water valves have opened to 100% which triggers a Nagios alarm. This created an interesting initial and ongoing investigation and discovery. Each Liebert XDP typically cools five Cray XK7 cabinets – some only four. Using the mapping of cabinets to XDPs and the timestamps of the alarms, particular jobs quickly emerged that were using the majority of the nodes served by the XDPs that alarmed. Table I reflects the alarms and jobs.

Table I. XDP Valve 100%

Start and End Times	Alarms	Code
03/24/2014 05:38:53 03/24/2014 09:53:03	6	DCA_GPU
03/28/2014 00:47:56 03/28/2014 01:29:29	4	DCA_GPU
04/18/2014 11:19:00 04/18/2014 17:34:54	5	NLN.CUDAXK7-MPI

Given our history with DCA++ [6], it was no surprise to find the GPU version of this code stressing the system. The DCA++ code won the Gordon Bell Prize two years in a row, so efficient computation resulting in high loads is expected. Of course, stress was the logical conclusion once realizing which code was running on the cabinets being served by the alarming XDPs, but before that, there was obviously potential for some situation in the machine room or other problem with the alarming XDPs. Using a graph from MRTG portrayed in Figure 4, it was again easy at a glance to spot the cadence of the peak computations in the code and the alarms from Table I.

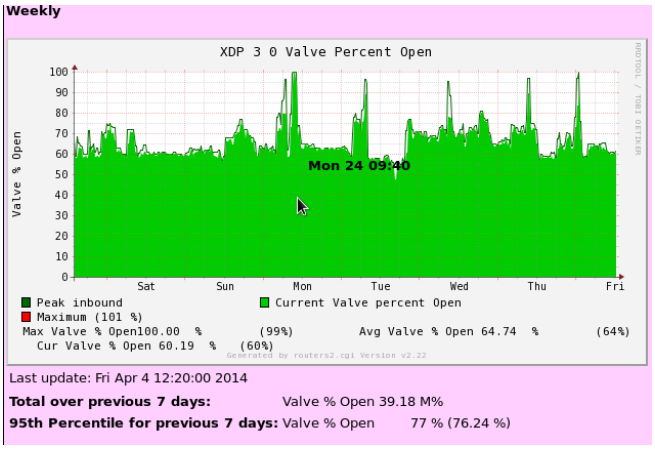

Figure 4. XDP 3_0 Valve Percent Open

Given the reaction of the cooling system, it seemed it might be interesting to see the power consumption for the code. Figure 5 provides an overall graph of the power response to the code.

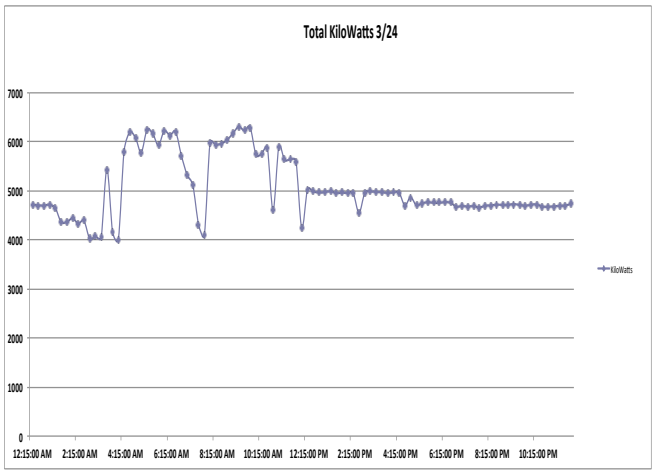

Figure 5. Total OLCF Power

Between 9:15AM and 9:45AM, the peak power usage was realized hovering around 6.3 Megawatts. Note that one of the 100% valve openings occurred at 9:40AM in Figure 4. As should be expected, there is a direct correlation of power consumed to the reaction of the chilled water valve.

Using another graph in Figure 6, the load provided by the other code in Table I called NLN.CUDAXK7-MPI is easily spotted across all XDPs. This illustrates the reaction of the entire cooling system to a very efficient run. When investigating this code, it was determined that the user had recently reported a 2X speedup in his code after improving the GPU efficiency of a tree traversal algorithm. Once again, the graphs provide a nice visual to see the reaction of the cooling system.

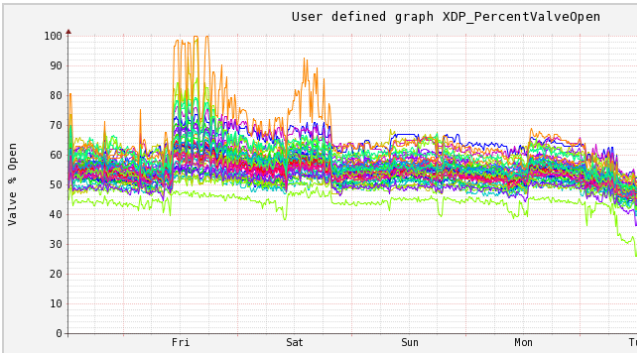

Figure 6. XDP Valve Percent Open for NLN Run

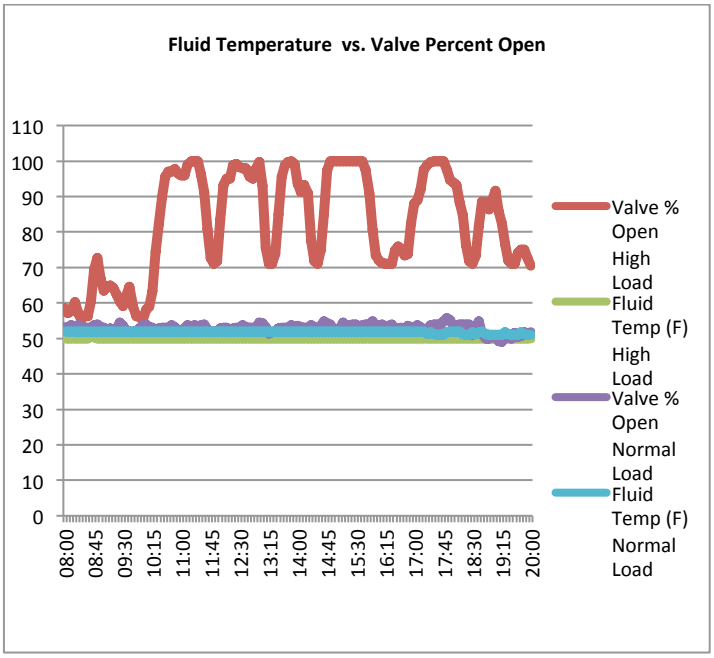

Figure 7. Fluid Temperature vs. Valve Percent Open

As mentioned earlier, often more than one data point is required to determine if there is actually a real problem that needs addressing. In the case of the valve opening completely, that is simply a reaction to load and would only become a problem if there was a subsequent rise in refrigerant temperature. This circumstance would indicate that the maximum amount of chilled water was unable to continue cooling the refrigerant, which is used to cool the system. Figure 7 provides a chart that shows both the chilled water valve reaction to NLN.CUDAXK7-MPI on a particular XDP and a plot of the refrigerant temperature on that XDP during the same timeframe. Given the refrigerant temperature did not stray from a flat line of fifty the entire time, it's easy to conclude that the chilled water was able to continue keeping the system cool. The plots of a typical load provide another interesting observation in the same chart. Notice that the fluid or refrigerant temperature under normal load is actually about two degrees higher. This seems to imply that not only did the valve opening to 100 percent continue keeping the system cool but also actually dropped the refrigerant temperature by two degrees. This might indicate that the system is actually overreacting to the load.

Following the possibility of some of the XDPs overreacting to increased load, there have so far only been two XDPs that alarm at 100 percent during high loads. During all three incidents outlined in Table I, only two XDPs alarmed, and it was the same two for each incident. Referring back to Figure 6, clearly all XDPs respond to the load, but the 100 percent responses only come from XDP 6_4 and XDP 3_0 indicated respectively by the spiking orange and obscure yellow behind the orange. These two XDPs are not in close proximity and frankly sit on opposite sides of the room four rows apart. The alarms triggered the investigation and discovery, so that was certainly a positive outcome, but investigation into why only those two are generating extreme circumstances continues.

VI. Conclusion

In summary, the new Cray cooling monitoring system is serving several purposes for the center. While it provides alarms for events that can highlight issues that need to be addressed, another welcome outcome has been the discovery of very compute intensive and efficient codes running at OLCF. Accounting can provide the quantity of a compute resource a particular code uses, but it cannot provide trends of infrastructure needs or issues that can be seen when particular codes are running. Alarms provide the trigger for discovery that can lead to investigation into power requirements for codes and future systems. Future work will be completed to harvest cooling data for the XC30 and incorporate it into the production monitoring system.

REFERENCES

[1] "MRTG"

[2] "QuickServer Gateway"

[3] "Nagios"

[4] "BACpypes"

[5] "RRDtool"

[6] M. Summers, "DCA++: Winning the Gordon Bell Prize with Generic Programming," Cray User Group Conference 2009