SAND2009-6071C

Resource Health Characterizations for Interactive and Autonomous Proactive System Administration and Scheduling Decisions

Jim Brandt, Frank Chen, Vincent De Sapio, Ann Gentile, Jackson Mayo,
Philippe Pébay, Diana Roe, David Thompson, and Matthew Wong
Sandia National Laboratories
MS 9159, P.O. Box 969
Livermore, CA 94551 U.S.A
ovis@sandia.gov

ABSTRACT

As high performance computing (HPC) clusters continue to grow in size and complexity and to offer an ever increasing array of services, the tools available to system administrators for efficiently utilizing and troubleshooting these resources has remained relatively stagnant. Though there are a plethora of tools and utilities available for managing various aspects of this, they are disjoint and require the interested and proficient system administrator to write one off scripts for culling the relevant data from the appropriate log and database files and making sense of it. This is typically done as an administrator gains experience with a system, develops hunches about probable causes and has the time to chase down supporting evidence.

Sandia National Laboratories hosts a variety of large HPC resources in support of its various programs and hence has a keen interest in their efficient and stable operation. To this end, we have been investigating the causal relationships leading to instability and failure in such systems and developing methodologies for quantifying the state of health of the resources based on these relationships. Such quantification can in turn be used to make scheduling decisions and/or invoke automated responses.

In pursuit of this goal we have developed a tool (OVIS) for scalable collection, analysis, and visualization of the large amount and variety of information that is or can be collected on such systems. Central to its design is the ability to handle the large amounts of data and to enable the fusion of information in various forms from disparate sources necessary to perform the resource state characterizations. This tool can also facilitate efficient administration of large systems by providing a detailed system-wide view.

This poster presents our preliminary work in including the fusion of information from hardware, system, resource manager (SLURM), and system log files to establish resource health characterizations as well as how these run-time characterizations can drive automated response through interaction between OVIS, SLURM, and system resources.

1. INTRODUCTION

As compute clusters increase in size and complexity, the overall mean time to failure (MTTF) necessarily decreases as a result of increased component count but stable component level MTTF. This makes timely recognition of failures or accurate prediction of impending failures increasingly important in order to provide meaningful invocation of response or defensive mechanisms. Such recognition or prediction is difficult as signatures of such conditions are largely unknown and discovery of them is contingent upon efficent analysis of large amounts of temporal and spatial system data residing in various forms, files, and databases. While this level of data can be obtained from compute clusters, it does not innately present itself in such a way as to support such analysis. There are currently no standard tools to relate textual data in one context to that in another or to numeric data in yet another. Tools that typically collect state data may keep this data for only limited times or at limited fidelity, neither of which will enable detailed analysis. Finally, such tools do not innately incorporate sophisticated tools for analysis, which adversely impacts the possible timescales for response to the analysis results.

We have been developing OVIS [1] an open-source tool for large-scale data collection, visualization, and analysis of large HPC cluster data. Central to its design [5] is the ability to collect, store, and analyze large amounts of state data from a variety of data sources and formats. The analyses are used to determine various characterizations of resources which are intended to, in turn, drive a variety of responses from initial resource allocation to automated resource replacement in an existing allocation pool [4].

In this poster, we will discuss our continuing developmental efforts on OVIS, how they faciliate determination of health characterizations, and how such characterizations can be used in driving administrative response and scheduling decisions that can be performed either interactively or autonomously to be performed by the system administrator or resource manager. We present initial work in a proof-of-concept interaction between OVIS and the resource manager, SLURM[2, 7], both for determining the characterizations and for invoking response based upon them. Material in this document will be augmented in an accompanying presentation in the poster session demonstrating determination of resource health characterizations analytically using data obtained on a production cluster at Sandia National Laboratories.

2. DETERMINING RESOURCE HEALTH CHARACTERIZATIONS

Characterization of the state of health of cluster resources depends upon the ability to recognize normal and significant

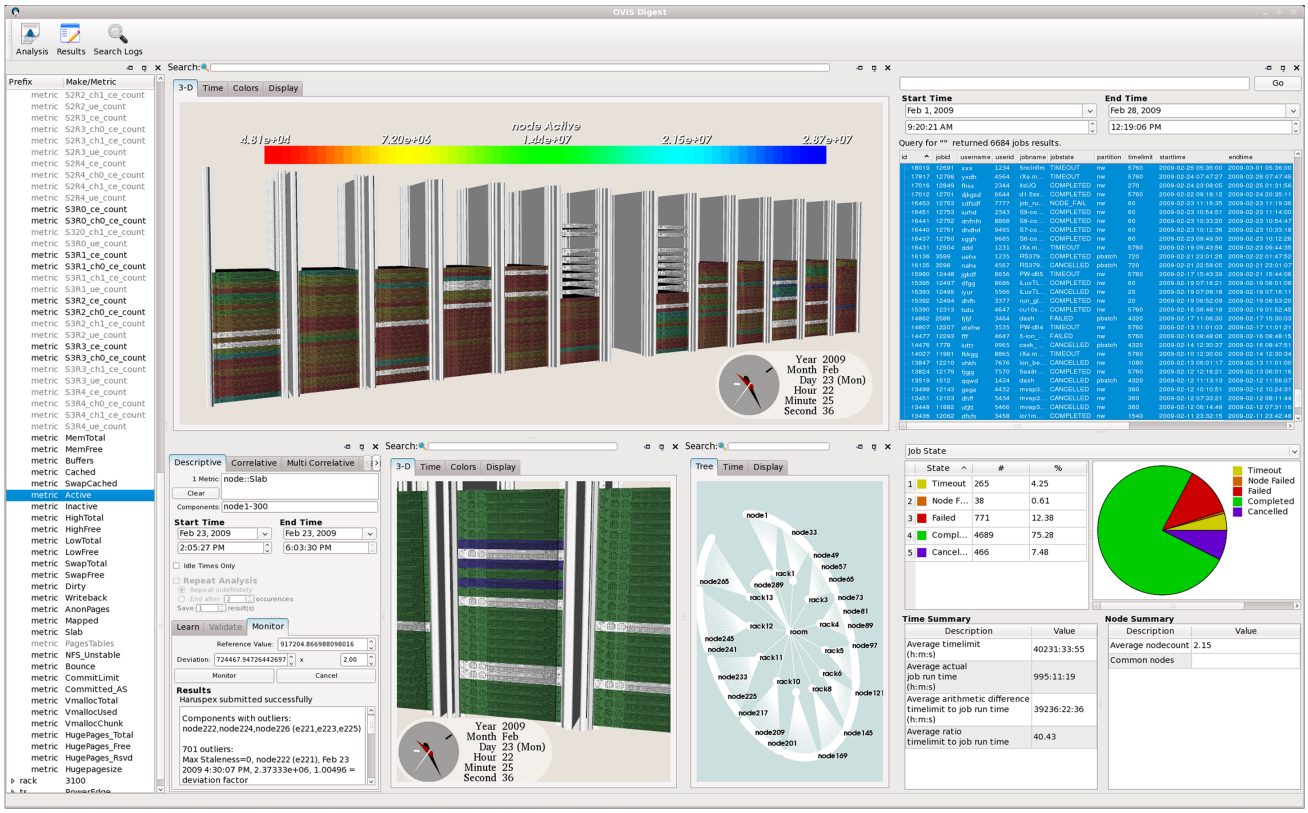

Figure 1: The OVIS GUI displaying geometrically correct physical 3-D representations of the cluster, colored by raw values and by probability relative to a calculated model; searchable scheduler and system log information; and analysis engines. System metrics are dragged onto either the physical view or the analysis box for display or processing. OVIS enables integration of information in order to determine numerical characterizations of resource health that can be used to drive autonomous response and scheduling decisions.

abnormal state. These are non-trivial tasks, as indicators for normal and abnormal behavior may be unknown and may be difficult to distinguish, given complex interdependencies of many state variables. Furthermore, the need for run-time data collection and processing of state data requires sigificant infrastructure capabilties. Finally, mere abnormality does not guarentee impending failure.

We have developed a tool, OVIS, for scalable collection, analysis, and visualization of the large amount and variety of information that is or can be collected on large HPC Clusters. Examples include numeric component level data such as voltages, temperatures, and application queue/run times as well as textual data such as syslog error names, application names, and the identities of resources involved in an application run. The OVIS interface, which consists of interacting components to provide visual, numeric, and textual analysis of such data is shown in Figure 1.

OVIS's analysis capabilities are principally targeted toward outlier detection in accordance with our hypothesis that statistically aberrant manifestations of values in the system could be used as an early predictor of impending failure. One of the main research issues then is to determine what sets, if any, of such behaviors are indicative of what types of health issues or potential failures. Using OVIS's analysis tools, both single and multi-variable statistical models can be derived and probabilistic descriptions of outlier behavior relative to the models obtained. The simplest example of this is a single variable case, an example of which is shown in Figure 2 [6]; outliers are displayed both in textual (l) and visual (r) form.

In order to associate failed jobs and system error conditions with behaviors of sets of associated system state metric values, recent work in OVIS has included a pilot implementation of an integrated capability to search the SLURM job databases and system/console logs. Due to SLURM's scheduling capabilities, the SLURM databases can keep detailed records on, among other things, job exit state and start and end times. This information can be used to investigate relationships of jobs with system data values evinced in the system. Further evidence of system errors can be obtained from system logs (e.g., [3]). Although these tend to be written in textual form and stored in files, we have written a prototype capability to convert such information from its native format on our system into SLURM's database schema.

Figure 3 shows this new capability in which SLURM job and system error log data can be searched upon (l) and used to automatically invoke an analysis (r). In this simple example the relationship between out-of-memory (OOM) events initially recorded in the console and system logs and failed

jobs is being investigated taking into consideration proximity in time and nodes in common. Once it is discovered that failed jobs and OOM events occur relatively close in time, the user can instantiate an analysis of active memory around the time of interest on the nodes of interest. Characterizations of memory utilization and detection of outliers relative to normal behavior are determined. If outlier behavior in memory can be correlated with the failed job and the OOM events in the logs, then such behavior can be used as health characteristic, with degree of probability used to determine a numerical indication of health.

3. UTILIZING RESOURCE HEALTH CHARACTERIZATIONS TO TRIGGER AUTONOMOUS RESPONSE

Expressing resource health in probabilistic terms enables early autonomous response. For instance, actions may be automatically taken when a value evinced in the system has less than a certain probability, given the other values that evinced in the system. Detection of less probable events can indicate a significant condition early than than raw data value thresholds which must be set at more extreme values in order to minimize false positives and typically do not take into account environmental conditions. A numerical representation faciliates autonomous triggering, in a way that more complex descriptions cannot. However, note that for complex system relationships or for drastic responses, system-administrator driven invocation may be preferred. Additionally, some issues, such as physical hot spots, can perhaps be more easily determined by the human eye processing the visual display than by numerical analysis.

3.1 Using Resource Health Characterizations to Drive Resource Allocation

We intend for resource health characterizations to be used not only to drive autonomous invocation of defensive mechanisms, but also to drive resource allocation.

Implementation of this requires us to 1) obtain application resource requirements, 2) determine resource characterizations 3) determine resources to allocate that fit the conditions, and 4) integrate this information with the scheduler at allocation time. The resource manager, SLURM, supports Resource Selection Plugins which are intended to determine the resources to be allocated for a given job, given a representation for a job's scheduling requirement specification. We plan to implement within such a plugin interaction with OVIS in which the resources which are allocated are determined based on the resource health characterizations.

OVIS would be used in a continous monitoring and characterization mode in order to have up-to-date resource characterizations available whenever the node allocation is required. An interaction between the two tools is more desirable than full integration of the tools into a single intelligent resource manager as the resource manager should not responsible for large-scale data collection and sophisticated analysis capabilities. While SLURM does support Job Accounting Gathering Plugins for obtaining data from verious sources, for example, `/proc`, the SLURM storage was not designed for long-term high-fidelity data storage of this type nor for representations designed with analysis as their primarly goal.

Note that the resources chosen for allocation may not be the most healthy resources, as the most healthy resources may be more optimally saved for the most important and/or long term runs; less healthy resources might be allocated for less important or short term jobs. This type of criteria can be obtained from the job submission information where a time limit is often required. If there are not enough resources available to satisfy a request, the job may be delayed until such resources become available. In the case of a high priority, long lived, or computationally intensive job, the delay in start time may be preferable to the delay in run time of a job with extensive checkpointing implemented as a defensive mechanism. Accurate resource health characterizations could be used to approximate the mean time to failure of individual components and hence maximize the checkpointing interval through appropriate resource selection. Currently the resource manager has no way of making such determinations.

4. FUTURE WORK

While developing the interaction mechanisms described here, continuing work in the OVIS projects involves investigation of failure indicators and failure modes, incorporation of additional data sources and infrastructure to support such, development of additional analyses that can be used to determine resource characterizations, and development of autonomously invoked warning indicators.

5. ACKNOWLEDGMENTS

The authors would like to thank Danny Auble and Moe Jette of Lawrence Livermore National Laboratories for useful discussions related to their SLURM project.

The authors were supported by the United States Department of Energy, Office of Defense Programs. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

6. REFERENCES

[1] OVIS. see `http://ovis.ca.sandia.gov` and references therein.

[2] SLURM: A Highly Scalable Resource Manager. `https://computing.llnl.gov/linux/slurm`.

[3] syslog-ng. `http://www.balabit.com`.

[4] J. Brandt, B. Debusschere, A. Gentile, J. Mayo, P. Pébay, D. Thompson, and M. Wong. Using probabilistic characterization to reduce runtime faults on hpc systems. In *Proc. of the 8th IEEE Symposium on Cluster Computing and the Grid (Workshop on Resiliency in High-Performance Computing)*, Lyon, France, May 2008.

[5] J. Brandt, A. Gentile, B. Debusschere, J. Mayo, P. Pebay, D. Thompson, and M. Wong. OVIS 2: A robust distributed architecture for scalable RAS. In *Proc. of the 22nd IEEE International Parallel & Distributed Processing Symposium (4th Workshop on System Management Techniques, Processes, and Services)*, Miami, FL, Apr. 2008.

[6] J. Brandt, A. Gentile, J. Mayo, P. Pébay, D. Roe, D. Thompson, and M. Wong. *OVIS 2 User's Guide*. Sandia National Laboratories, Livermore, CA, 2009.

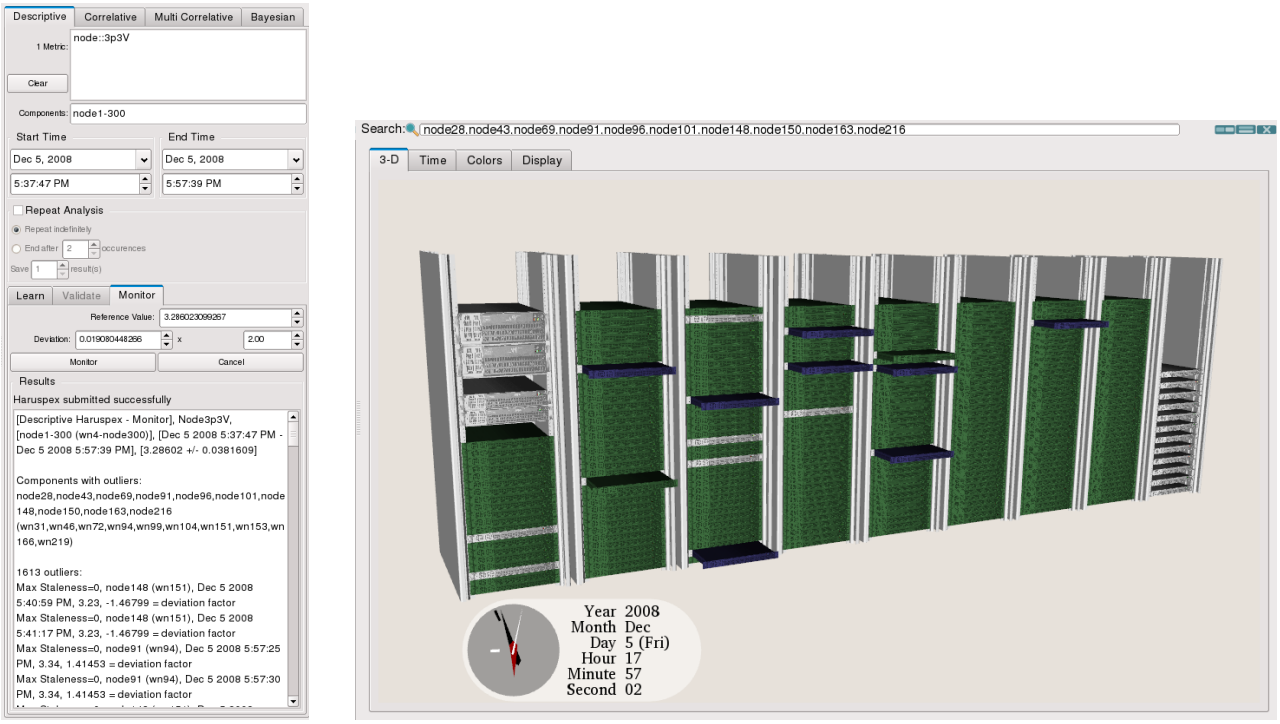

Figure 2: Outlier analysis results presented in textual (l) and visual (r) form. Nodes whose values are within the specified probabilistic tolerance are colored in green, outliers whose values are above/below the model are shown in red/blue.

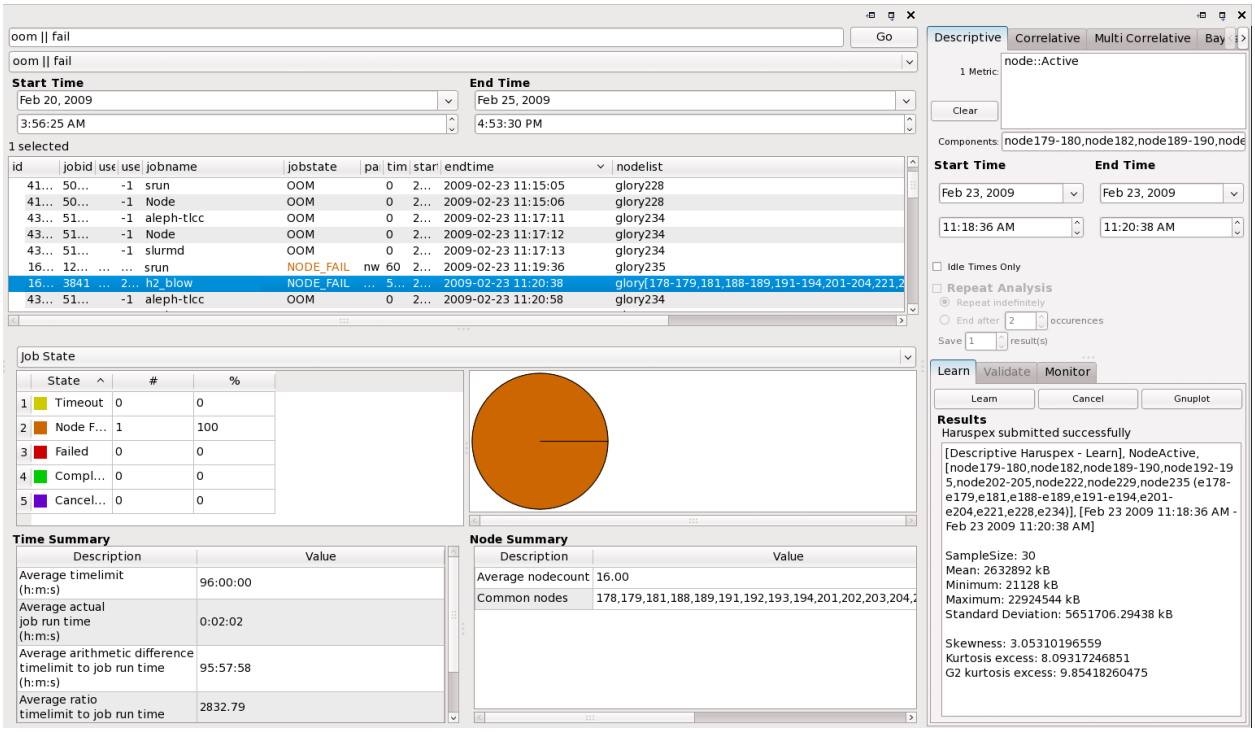

Figure 3: Intergrated capabilities for searching SLURM job logs and system error logs can be used to determine possible causes of failures. Analysis panes are automatically populated with job nodes and time in order to determine the relationships between statstical abnormalities with job and system state. Here the relationship between a failed job, an out-of-memory system error, and job memory statistics is being investigated.

[7] A. Yoo, M. Jette, and M. Grondona. Slurm: Simple linux utility for resource management. *Job Scheduling Strategies for Parallel Processing*, 2862:44–60, 2003.