# SANDIA REPORT

SAND2005-4558
Unlimited Release
Printed July 2005

# Meaningful Statistical Analysis of Large Computational Clusters

J. M. Brandt, A. C. Gentile, Y. M. Marzouk, and P. P. Pébay

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

| | |
|---|---|
| Telephone: | (865) 576-8401 |
| Facsimile: | (865) 576-5728 |
| E-Mail: | reports@adonis.osti.gov |
| Online ordering: | http://www.doe.gov/bridge |

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

| | |
|---|---|
| Telephone: | (800) 553-6847 |
| Facsimile: | (703) 605-6900 |
| E-Mail: | orders@ntis.fedworld.gov |
| Online ordering: | http://www.ntis.gov/ordering.htm |

SAND2005-4558
Unlimited Release
Printed July 2005

# Meaningful Statistical Analysis of Large Computational Clusters

J. M. Brandt
Sandia National Laboratories
M.S. 9152, P.O. Box 969
Livermore, CA 94550, U.S.A.
brandt@ca.sandia.gov

A. C. Gentile
Sandia National Laboratories
M.S. 9152, P.O. Box 969
Livermore, CA 94550, U.S.A.
gentile@ca.sandia.gov

Y. M. Marzouk
Sandia National Laboratories
M.S. 9051, P.O. Box 969
Livermore, CA 94550, U.S.A.
ymarzou@sandia.gov

P. P. Pébay
Sandia National Laboratories
M.S. 9051, P.O. Box 969
Livermore, CA 94550, U.S.A.
pppebay@ca.sandia.gov

**Abstract**

Effective monitoring of large computational clusters demands the analysis of a vast amount of raw data from a large number of machines. The fundamental interactions of the system are not, however, well-defined, making it difficult to draw meaningful conclusions from this data, even if one were able to efficiently handle and process it. In this paper we show that computational clusters, because they are comprised of a large number of identical machines, behave in a statistically meaningful fashion. We therefore can employ normal statistical methods to derive information about individual systems and their environment and to detect problems sooner than with traditional mechanisms. We discuss design details necessary to use these methods on a large system in a timely and low-impact fashion.

# Acknowledgement

The authors would like to thank Dave Evensky and David Thompson (Sandia) for technical advice.

# Contents



# Figures

This page left intentionally blank

# Meaningful Statistical Analysis of Large Computational Clusters

## 1 Introduction

Modern computational clusters consist of large numbers of machines containing sensors from which data about their physical states (*e.g.*, temperatures, fan speeds, voltages) can be obtained. While this data is easily accessible on a per-machine basis, it is hard to derive meaningful information from because it is hard to manage and because the interactions among machines and between the machines and their environment are not well-defined.

Current cluster monitoring tools, such as Ganglia [1] and Supermon [5], efficiently gather the raw information, but they do not provide automated analysis and mainly present administrators with the ability to view the primitive data on a per-node basis. Typical automated system management based on this type of data (for example in tools from IBM [3] and HP [2]) consists of comparing the instantaneous values of the data on a per-machine basis to predefined thresholds and either sending notification to the system administrator or automatically shutting down or rebooting the system in response. This threshold reflects an extreme case for which nodal failure is expected to be imminent and therefore means that a problem detected in this fashion is detected only when the problem has finally become severe. Furthermore, the reason the threshold was crossed is, in general, never known unless the same machine cycles frequently enough to bother someone into taking it in for repair.

We are attempting to make automated meaningful analysis of this data in the absence of a fundamental understanding of the total internal state of the nodes, the interactions between nodes, and the interaction of machines with their environment. Typically, clusters consist of a large number of identical systems that should behave similarly under identical circumstances. The cluster can be viewed as a set of ensembles (*i.e.*, meaningful groups of nodes such as those running the same job or located in the same area) that should behave in a statistically meaningful fashion. This allows us, to first order, to use normal statistical methods to derive information about individual systems as well as the entire system and its environment. Rather than manage the cluster on a node-by-node basis, we are bringing statistical understanding to the world of cluster management.

In this paper we describe some statistical methods for useful analysis of environmental data taken from a large number of identical systems housed in close proximity. We concentrate on thermal data as it is directly related to CPU activity and cooling, but these methods can just as well be applied to power supply voltages, memory error rates, and other diagnostics. We consider the scope and limitations of our statistical methods and present real-life cases demonstrating that meaningful data about a cluster can be obtained in a statistical fashion. We describe design issues relevant to implementing statistical methodologies on a large system in a timely and low-impact fashion. We conclude with future directions.

# 2 Scope of the analytical regime

High performance cluster computers are typically comprised of many identical server-class multi-processor machines. This homogeneity lends itself nicely to statistical analysis as it is expected that, given the same environment (*e.g.*, air temperature, computational load), the behaviors of the machines' physical parameters (*e.g.*, temperatures, fan speed, voltages) should follow some reasonable distribution. (While heterogeneity does exist in grids [6], they are comprised of federations of clusters which are often locally-managed homogeneous resources.)

To first order, then, we can consider statistics of ensembles such as groups of nodes running the same job, or those geographically co-located. We look at both static and time histories of the statistical distributions. Statistical deviations in behavior are a reflection of abnormality in the environment or in the system. An example of an abnormality in the environment is a faulty setup where hot air from one machine's exhaust is being directed into the intake of another machine. An example of abnormality in the system is a poorly attached heat sink in one node that causes heat to dissipate less efficiently in that node. Either type of abnormality is significant as it can have impact on the longevity of the system.

There are several potential difficulties to a purely statistical approach. The first difficulty is that once these seemingly identical machines are configured into a closely co-located cluster, they necessarily experience *some* non-uniformity of environment and, further, can potentially affect one another. For instance, we expect that the ambient temperature experienced by any two machines will vary because the airflow within a room or rack is generally non-uniform. However, unless the environment is non-uniform in a consistent way that can be subtracted out, we risk that the skew due to this environmental field overwhelms the level of detail necessary to resolve normal statistical deviations. Design flaws in the system could not only enhance non-uniformity, but also may correlate a subset of the system so that the independence assumption necessary to many statistics is invalidated. Also, secondary correlations between independent jobs can arise through competition for shared resources. Finally, while equipment must meet manufacturer standards, we don't know that the variations in individual nodes would follow a normal distribution, even in purely isolated conditions.

With no thorough understanding of the detailed interactions between nodes and between the nodes and their environment, there is no way to officially resolve these difficulties. Empirically, however, we observe no gross abnormalities in the temperature distribution across the cluster that would suggest that non-uniformities in the environment are the primary factors determining the distribution. Expected factors such as computational load affect the cluster in an intuitively reasonable fashion; members of a job group (whose identities are available from the scheduler responsible for placing the job on the nodes) behave very similarly (§3.1) during an application's run.

Therefore, while we are at liberty to consider the statistics of rational ensembles, in light of the caveats above we risk drawing either false negative conclusions or false positive conclusions. However, if we draw false negative conclusions (*i.e.*, miss a problem in the cluster), we are no worse off than we were with the thresholding mechanism. If we draw a false positive conclusion (*i.e.*, flag a correct distribution as faulty), we can avoid acting on it unless it is something that we can rationally verify. Note that the state-of-the-art does not set the bar for improvement in analysis techniques particularly high. Typical automated cluster management tools flag nodes as being faulty in their temperature when their values cross a threshold value, roughly around 65°C.

# 3 Case studies

In this section we present examples of what we have observed in statistical thermal analysis of ensembles and what meaning can be derived from them. This includes problem discovery of individual nodes, problem discovery in cluster design, discovery of global environmental effects, avoidance of drawing faulty conclusions about problems, and modeling environmental effects on temperature distributions to allow for baseline subtraction of these effects.

A block diagram of our cluster appears in Fig. 1. It is the design of each rack as it came from the manufacturer. Nodes are mounted in vertical racks, and we refer to horizontal cross-cuts as slices. A node's air intake is in the front and exhaust is through the back (mesh doors exist on rack fronts and backs). Around the middle of each rack is a largely empty gap, with only a small piece of non-compute node gear which does not fill in the gap. Air enters the room through perforated floor tiles and exhausts through grates in the ceiling. Among other variables, temperature and fan speed are obtainable on a per processor basis.

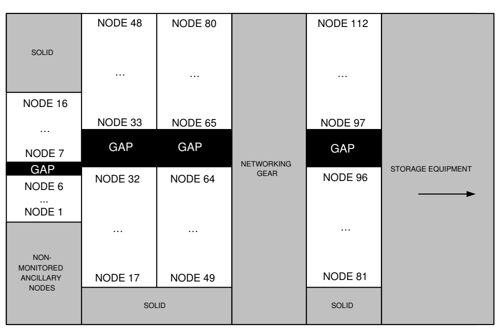


**Figure 1.** Block diagram of our cluster (not to scale). Around the center of each rack is an empty gap.

Temperature is reported in integer degrees. The fan has a few discrete states. A control system automatically raises fan speed in response to high CPU temperatures.

### 3.1 Job ensemble: discovery of an abnormal node

First, we consider a small test cluster of 4 nodes to which we have complete access. While its size limits the rigor of statistical conclusions that can be drawn, it does allow us to control the the environment of the nodes. This case has the added benefit of illustrating typical behavior on cluster nodes during a job.

Figure 2, top left, shows the temperature history of the four nodes in a test run. All nodes are initially idle. When a job is started on the nodes the temperature increases fairly rapidly, then plateaus. When the job ends the temperatures return to their idle values. Some variation is found in the values of the nodes, but in general nodes in the same job group move in thermal concert with each other as typically they are doing the same types of tasks in parallel.

We then deliberately altered one node, loosening the thermal bond between the processor and its heat pipe, causing it to dissipate heat less efficiently. Figure 2, top right, shows the time history for the same job in the altered case. When nodes were idle the altered node was not meaningfully different than other nodes in its job ensemble. However, when a job started on the faulty node, heating up the node, the node's thermal profile rose higher and faster than those of its peers. This behavior can be detected by comparing simple statistics for the jobs. In Fig. 2, bottom, we plot the mean and standard deviations of the temperatures of the nodes in the job group in both cases. In the altered case we see that the mean job group temperature and standard deviation is greater than that of the normal case, and that the deviation is greater during the job run, when there is more heat to be dissipated, than in the idle stage.

From this example we see that

1. in a well-behaved environment, detection of the faulty node can occur at temperatures lower than traditional threshold values, and
2. some problems may only manifest themselves, or at least more strongly manifest themselves, under specific conditions, such as increased load.

Together this indicates that one must be careful to consider ensembles under a range of conditions. In the remaining cases, then, we consider statistics with the added complication of a real-world environment.

### 3.2 Geographic ensembles: a fixable abnormal environment

In this case we examined the average fan speed of nodes under production conditions in our cluster. We anticipated that there could be great variation in fan speed, reflecting reaction to the changing load in the cluster. However, a histogram (unshown) of the distribution of fan

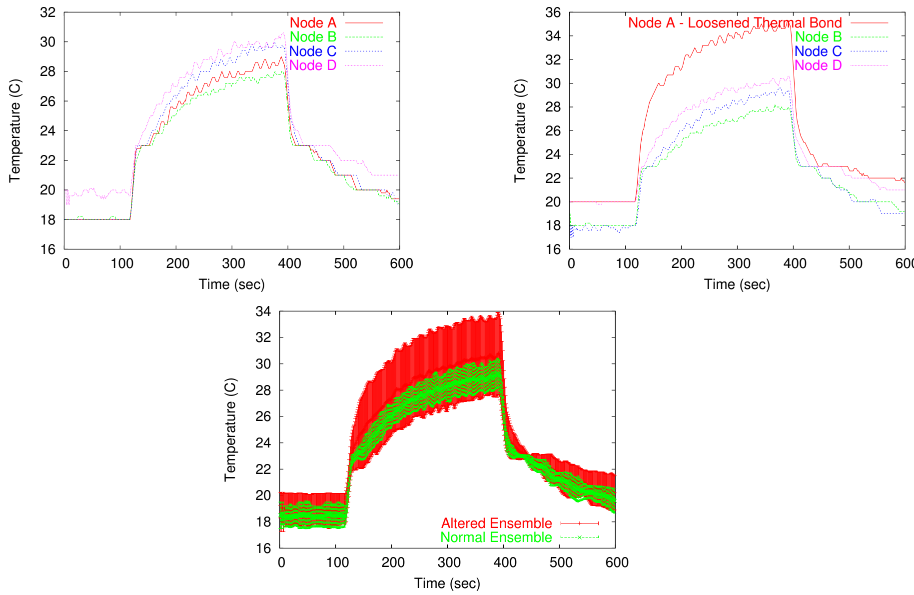


**Figure 2.** Top: thermal behavior of individual nodes in a job ensemble in a controlled test cluster. Temperatures are low at idle, rise when a job is running, and drop back down when the job is completed. There are variations in individual nodes, but they have generally similar behaviors (left). When the thermal bond of the heat pipe of a node is deliberately loosened (right), that node exhibits meaningful thermal variation from the group when under load. Bottom: job ensemble statistics of the normal and altered cases. Mean ensemble temperature and standard deviation are greater in the altered case. The deviation is greater under load.

speeds in the cluster showed that the vast majority of nodes exhibited the same stable fan speed over time, independent of load, with only a few nodes exhibiting different behaviors.

We considered several geographic ensembles, expecting the discordant values to be located either closest to the ceiling or closely co-located indicating a hot spot. While we cannot prevent heat rising, a hot spot may indicate a cluster design flaw or a problem in room layout that affects the airflow in a consistent fashion.

Consideration of the average fan speed as a function of height showed that fan speed averages in the rows bordering the gaps in the rack were the highest. Examination of the racks showed that the basic physical cluster design was flawed in that hot exhaust air from the back of the racks was being recirculated through the gap in the center of the racks into the intake of the machines. Blocking this gap in 3 of the 4 racks (keeping the 4th rack un-blocked for comparison) caused the fan speeds of the nodes above the break in the 3 racks to fall in line with the values of the rest of the cluster.

In this case a correctable problem was discovered. We emphasize that this condition was not even hinted at by the traditional per-node threshold-based monitoring supplied with the cluster.

## 3.3 Correlating independent job ensembles: global effects

Despite the case discussed in § 3.2, we know that there will be cases where the environmental field affects the system in a way that is uncorrectable, though we hope not undetectable. Here we consider the correlation between the time averaged values of the temperatures of two different job groups. While the long term temperature behavior of a job group is typically relatively stable, we found a time-frame in which the temperature of each job group rose and fell a few degrees in concert with each other (Fig. 3 left). Given the independence of the jobs this suggests that the change is due to some global factor. The size and time period of the change (a few degrees over an hour) are consistent with a drift in room temperature.

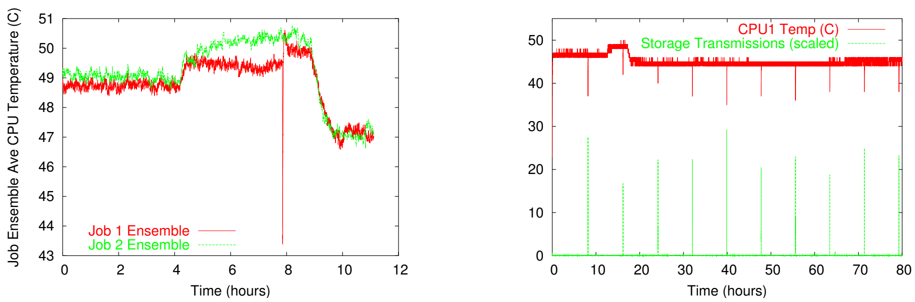


**Figure 3.** Job Ensembled Averaged CPU Temperature during a portion of two simultaneously running jobs (left). Cross-correlation in the thermal behaviors of independent jobs indicates global environmental factor affecting the cluster. Temperature drops in job #1 coincide with data write-outs (shown for one node in the ensemble)(right). Data was taken at 2-second intervals. Job ensembles consist of 32 and 14 nodes, respectively.

Cross-correlation of jobs in this fashion allows one to grossly account for environmental conditions like these, although it may require the complexity of handling slight offsets in time. It can also bring one's attention to potential changes in the environmental configuration that can have large ramifications, such as the relocation of another piece of equipment that alters airflow.

## 3.4 Correlating job ensembles: avoiding false positives

Some behaviors may seem to be inexplicable and hence reflective of a problem. For instance Fig. 3, left, also shows a large temperature decrease at (at time around 8 hours) in job #1. Cross-correlation with the other job indicates that this is not a global cluster effect, unlike the temperature rise in § 3.3. Ensemble examination shows that this happens to all nodes involved in the job, which, from scheduler data are widely distributed throughout the cluster, making a nodal problem unlikely. In this case, we were forced to consider additional variables which might have secondary effects on temperature. Correlation with transmission data shows that the temperature decrease is not a problem, but a reflection of a normal phenomenon: computation stops while the job writes data to disk.

## 3.5 Geographic ensembles: unavoidable environmental field

This case is the temperature analogue of the case described in §3.2. Once we corrected the airflow problem that affected the fans, we turned our attention to the temperature distribution, expecting that it would be more sensitive to the environmental field. Temperature

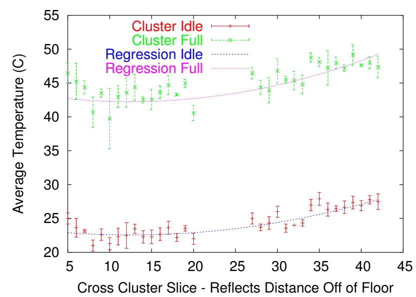


**Figure 4.** Average values of temperature of the nodes in a slice of the cluster shows the effects of the environmental field on the cluster. While temperature generally increases as a function of height, there is a region near the floor where the opposite is true. Increased load enhances the natural variation in individual nodes, seen particularly in slice 10. Temperatures were taken on a per-node basis every 2 seconds for slightly less than 0.5 hours in each case.

as a function of distance from the floor is shown in Fig. 4. In this case we see that while temperature does generally rise with height, there is a region next to the floor where the temperatures decrease with height. (Rack 1 is not included in the calculations or diagrams in this section since it does not contain nodes in this region - see Fig. 1). This initial decrease turns out to be due to warm air recirculating beneath the cabinet from the exhaust side back to the front due to the low pressure zone created by high velocity air emitted from the floor tiles. This skew within a completely homogeneous case surpasses the limit of what we are easily able to fix by tweaking the cluster design or making easy airflow alterations.

The question is: can we consider that skew as a subtractable baseline or will its existence mask other normal variations in the system? We consider the difference in this skew between the cases of all nodes running at idle and running the same computationally intensive code. To answer this question we attempt to find models to fit the data for these cases: a natural approach consists of modeling temperature as some function of height, multiplied by some random "noise" that is on average equal to 1. In fact, this noise factor is here to account for the parameters other than height that have an effect on temperature; there are evidently a great many of them, only considering manufacturing variations. A grasp at their relative importance with respect to height can be obtained by looking at the standard deviation of the noise – a standard deviation of zero would mean an exact fit to the polynomial approximation. Therefore, as a first cut, we decide to fit the large amount (both in time and in space) of observed data to the following model:

$$T \sim \mathcal{N}(P(h), \sigma), \tag{1}$$

where h and T respectively denote height and temperature, and $\mathcal{N}(m, \sigma)$ denotes the normal distribution with mean $P(h)$ and variance $\sigma$. In our case, $P$ is a polynomial function, and we tried the second-order case, *i.e.*, $P(h) = ah^2 + bh + c$ for which the coefficients $a$, $b$ and $c$ need be identified. With this relatively simple model, we are able to fit each of the idle and running data with a confidence level of up to 99%. We find variance is greater for the computationally intensive case which seems to emphasizes the physical discrepancies between computational nodes. Finally, the answer is: yes, we are able to model skew as a function of height thus reducing it to a subtractable baseline. Though we must use slightly different models for the idle vs. the computationally intensive case, the extra baggage required for this is just three coefficients and a variance. Figure 4 shows data and models for both the idle and the computationally intensive case.

We note that slice 10 exhibits a stronger difference in temperature range between its loaded and idle cases than the other slices. We have traced this back to node 86, which runs significantly cooler than its peers. This effect is greatly enhanced at higher computational loads. Upon investigation it turned out that this node's airflow is much greater than that of its peers even though the fan speed sensors claim its fans are running at the same speed as those of its peers. Further investigation showed that on this node fan speed is independent of what the speed sensor registers and indeed never changes. This explains the thermal discrepency observed for this node and is another example of an anomoly (in this case not driving the node toward thermal failure but perhaps fan failure) that is detected only through comparison of a nodes steady state variable's with those expected due to statistical analysis of a large number of similar nodes.

Finally we note that the ability to categorize the natural environment is valuable information in and of itself. Nodes running at consistently higher temperatures due to the environmental field may be susceptible to more problems, decreased performance, and shorter lifespan than those located elsewhere.

## 3.6 Summary

We saw that the valid regime for statistical analysis is sufficiently large that we can meaningfully investigate statistical distributions and detect problems earlier than the traditional per-node threshold-crossing mechanism. Additionally we can identify environmental issues that can be used for improving cluster design.

We note the posssibility of a point at which irregularities in the distribution due to unavoidable issues in the environment and cluster design could become dominant and overwhelm the ability to get *fine* detail information by purely statistical methods.

# 4 Implementation design considerations

The ability to do real-time comparisons of statistical data can be important in a facility where moves, changes and additions are commonplace, and seemingly small alterations can have large ramifications. We discuss design details necessary to use these methods on a large system in a timely and low-impact fashion.

Many, but not all, HPC clusters have environmental sensors on board as part of their design; these methods, of course, can only apply to the former. Since implementations of getting data from the sensors and even which elements are monitored vary from vendor to vendor, so far we have targeted only temperature and fan information which seem to be universal in monitored systems. Relationships among variables must be considered: fan speed and temperature are directly related and either taken separately can give fallacious results.

The thermal state of a node is significantly affected by its hosting of a running job. Large clusters support many concurrent jobs, so it is important to know the job to node mapping. Real-time implementation requires the ability to dynamically get this information from the scheduler, which one can do, for example in PBS [4], by simple additions to the prologue and epilogue scripts. Implementation of a post-processing version simply requires reading this information from the scheduler log files.

Scalability can be addressed by distributing the processing over a number of entities. Calculations can be independently done on separate ensembles. Larger ensembles can be represented by a dynamic random subset of peer nodes.

The impact of data collection on running applications derives from several possible sources. In-band collection requires one of the main processors executing an application to periodically collect and distribute data. The cycles required to do this are in essence stolen from the application. Data collection traffic may contend with application traffic for network resources. Solutions to these adverse effects are either out-of-band data collection where a separate processor is dedicated to collecting and distributing sensor information, or a separate data collection network. While it is preferable to do out-of-band data collection, this is not always an option. For the purpose of this work we are using in-band collection but notice less than 1% measured impact on running applications wall clock time while collecting five variables on two second intervals. This is a considerably greater rate of data collection than that used in traditional data gathering and monitoring tools.

In order to do correlations with historical data, one must also provide the ability to store and retrieve primary and derived run-time data. Depending on the statistics used, one can minimize storage by keeping only meta-data or gross statistics. A flexible archiving system can allow the user to keep shorter time histories in the active memory and if he wants to drill down into a particular situation he can call the data up from the archive.

# 5 Possible extensions

Ensemble statistics provide valuable insight into the behavior of a cluster, but these statistics are only the beginning of possible analyses that can be applied to the present data. Consider the interactions implicit among the time-dependent raw data collected from each node. It is clear that these raw values are conditioned on each other in a complex fashion. For instance, a high CPU temperature is "normal" when computational intensity is high, but may indicate a problem when the node is idle. If a node temperature drops suddenly, this may indicate a malfunction; if, however, node temperatures within the same job group all drop within a certain time skew of each other, this may indicate that the job is pausing for I/O. The acceptable time skew in this case is not clear *a priori*; rather, one would like this quantity to be learned, along with all the other conditional probabilities that represent the observed data.

We thus propose to build Bayesian networks that reflect these interdependencies, providing the user with a much finer and more automated description of conformance versus non-conformance. We also propose to use unsupervised learning algorithms (*e.g.*, clustering) to discover meaningful ensembles within the data. This capability may be useful when job scheduler information is unavailable, for instance; or when diagnosing a remote cluster for which the spatial co-location of nodes is unknown.

These analytical tools will not require information beyond that which is collected in the present work, and thus may be built atop the data collection framework described above.

# References


[1] Ganglia. `http://ganglia.sourceforge.net`.

[2] HP iLO. `http://h18004.www1.hp.com/products/servers/management/ilo`.

[3] IBM director upward integration. `http://www-1.ibm.com/servers/eserver/xseries/systems_management/xseries_sm.html`.

[4] Portable batch system (PBS). `http://www.openpbs.org`.

[5] Supermon. `http://supermon.sourceforge.net`.

[6] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. Supercomputing Applications*, 11(2):115–128, 1997.

# DISTRIBUTION:

1 MS 9051
Dawn Manley, 8351

1 MS 9152
Jerry Friesen, 8963

1 MS 0817
James Ang, 9224

1 MS 9151
James Handrock, 8900

1 MS 9158
Mitchel W. Sukalski, 8961

1 MS 9159
Heidi Ammerlahn, 8962

1 MS 9159
Mike Hardwick, 8964

1 MS 9011
John Howard, 8941

1 MS 9158
David A. Evensky, 8963

1 MS 9152
David C. Thompson, 8963

10 MS 9152
Jim M. Brandt, 8963

10 MS 9152
Ann C. Gentile, 8963

2 MS 9051
Youssef Marzouk, 8351

2 MS 9051
Philippe P. Pébay, 8351

3 MS 9018
Central Technical Files, 8940-1

1 MS 0899
Technical Library, 9616

1 MS 9021
Classification Office, 8511, for
Technical Library, MS 0899, 9616
DOE/OSTI via URL