OVIS: A Tool for Intelligent, Real-time Monitoring of Computational Clusters

J. M. Brandt, A. C. Gentile, D. J. Hale, and P. P. Pébay

Sandia National Laboratories, Livermore CA 94550 U.S.A.
{brandt,gentile,djhale,pppebay}@sandia.gov

Abstract

Traditional cluster monitoring approaches consider nodes in singleton, using manufacturer-specified extreme limits as thresholds for failure "prediction". We have developed a tool, OVIS, for monitoring and analysis of large computational platforms which, instead, uses a statistical approach to characterize single device behaviors from those of a large number of statistically similar devices.

*Baseline capabilities of OVIS include the visual display of deterministic information about state variables (*e.g.*, temperature, CPU utilization, fan speed) and their aggregate statistics. Visual consideration of the cluster as a comparative ensemble, rather than as singleton nodes, is an easy and useful method for tuning cluster configuration and determining effects of realtime changes.*

Additionally, OVIS incorporates a novel Bayesian inference scheme to dynamically infer models for the normal behavior of a system and to determine bounds on the probability of values evinced in the system. Individual node values that are unlikely given the current applicable model are flagged as aberrant. This can be a much earlier indicator of problems than waiting for the crossing of some threshold that is necessarily set high to preclude too many false alarms.

We present OVIS and discuss its applications in cluster configuration and environmental tuning and to abnormality and problem discovery in our production clusters.

keywords: cluster monitoring, Bayesian inference, RAS, abnormality detection

1. Introduction

Current monitoring of computational clusters of sizes ranging from tens to tens of thousands of nodes is typically performed in a simple fashion. First, get data via a push or pull mechanism from each node at an interval that allows new data to be available and the monitoring station to keep up. Second, apply a predefined rule set to the data collected on a node by node basis: if a threshold is crossed, apply the appropriate rule (*e.g.*, shut down a compute node).

Tools such as Ganglia [1] and Supermon [4] do the first efficiently, 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 management tools, such as those from IBM [3] and HP [2], compare these instantaneous data values on a per-machine basis to predefined thresholds and either send notification to the system administrator or automatically shut down or reboot the system in response. These thresholds reflect extreme cases for which nodal failure is expected to be imminent. Therefore, problems detected in this fashion are detected only when they have finally become severe.

While this basic methodology of considering singleton node values in light of gross rule sets is well suited to smaller clusters, we have found that a statistical approach to cluster monitoring and analysis can provide more meaningful information and enable earlier detection of problems. We have built a tool, OVIS, which uses simple statistical methods in order to facilitate gross fault detection and configuration analysis by visual inspection. More advanced methods, such as modeling via Bayesian inference, add a great deal of intelligence to the process, thus facilitating automated discovery of problems sooner than is possible

These 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. The authors would like to thank Y. Marzouk and D. Thompson for useful discussions.

1-4244-0054-6/06/$20.00

using static thresholds.

Our statistical approach to cluster analysis capitalizes on the fact that clusters 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 a normal distribution (while heterogeneity does exist in grids [5] they are comprised of federations of clusters which are often locally-managed homogeneous resources). Although the concept of taking advantage of the number and similarity of nodes in a cluster has been previously recognized [6], we know of no tool in practice that utilizes this for its fundamental methodology.

The baseline capabilities of OVIS include the visual display of raw data and their aggregate statistics. This method capitalizes on a human's ability to efficiently spot patterns and abnormalities. It is a quick-and-easy, non-computationally intensive way to gain the benefits of considering the cluster as a comparative ensemble, rather than as singleton nodes.

The Bayesian module increases capabilities, allowing the system to model out environmental effects that can't be changed, and subsequently enabling automatic detection of nodal aberration from normal behavior, where normal is defined as being within some administrator-defined probability bounds (*e.g.*, 95%) conditioned on the model inferred from the behaviors of a large number of statistically similar devices (peers).

We present an introduction to OVIS in Section 2. We discuss issues in the statistical approach due to non-uniform environments in Section 3. Methodologies and examples of problem discoveries in our production clusters are discussed for the visual and Bayesian approaches in Sections 4 and 5, respectively. We concentrate on temperature analyses in this work, as temperature is a well-known factor in equipment failure and airflow and cooling issues in machine rooms are a common problem. However, the techniques involved should be generally applicable to other variables such as voltages, memory error rates, etc. We conclude in Section 6.

2. Overview of OVIS

The centerpiece of the OVIS GUI is a physically representative display of node and cluster configuration, as illustrated in Figure 1. OVIS reads from an XML specified geometry file to build the display for a particular cluster's configuration.

Data, raw or derived, from either a saved data file or a direct data feed is overlaid, according to a user definable color scheme, onto the display. Encoding data values as colors which are mapped onto node positions on the display gives the user a very intuitive view of the data and how it relates to geography and node state. The color mappings can be customized but in general low values are mapped to bluer colors and higher values move toward the red end of the spectrum. There are options to map binned values to particular colors thus allowing easy discrimination between values falling near boundaries (e.g. 2 standard deviations from the mean boundary) or to produce a gradient display which facilitates better understanding of the distribution of values especially where spatial and temporal gradients are present. The gradient view can be put to good use in optimizing cold air distribution and choosing node groups and axis for application of Bayesian modeling techniques. The color scheme can be adjusted in real time to allow the user to get as fine-grained as the data will allow.

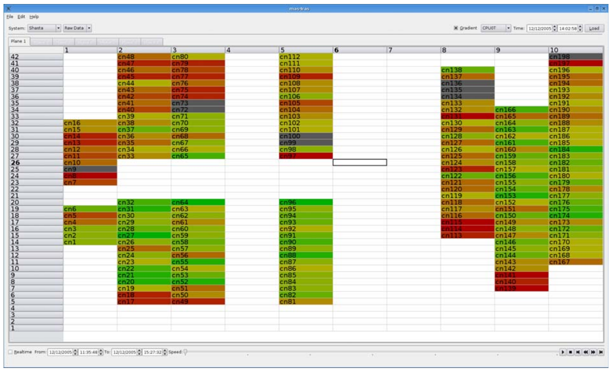


Figure 1. OVIS display showing run-time conditions and configuration of our Shasta cluster. Nodes are depicted in the physical layout of the cluster racks. Values of raw and derived quantities are displayed by color-coding of the nodes. Patterns and outliers are easily spotted by the eye.

Though the preferred method of data capture is out-of-band, OVIS provides a daemon-based in-band collection code that can be modified to suit any system and writes out data files in OVIS's native format. A translator from the Ganglia data format is also available.

Play, pause, fast-forward/reverse and go-to-end of file are available, allowing the user to review history or to go over an area with an adjusted color scheme or different variable display. A mouse-over feature allows immediate simultaneous viewing of concurrent variable values enabling the user to look for correlations among variables.

Modules available for doing statistical processing are (1) the baseline statistical module, which computes means and standard deviations across user defined groupings of nodes, and (2) the Bayesian module, which facilitates more advanced techniques such as modeling arbitrarily complex dependencies and computes probability bounds for variable values for each node given the model which applies to it.

3. Non-uniform Environments

Recognition of thermal outliers is difficult if not impossible in an environment where the computational loads and fan speeds of the nodes may vary greatly. Unfortunately, it is rare for uniformity to exist, so that, to first order, we can only consider statistics of ensembles of nodes in similar states, such as groups of nodes running the same job. These constraints can, however, be loosened somewhat if we can normalize temperatures, during stable periods, to effects such as CPU utilization and fan speed.

Normalization of temperature to account for such effects allows the user to visualize the cluster as a large set of statistically similar randomly placed devices, independent of what application may be running where. Departure from uniformity is then due either to single node anomalies or environmental effects. Environmental cases are distinguished by the extent and placement of the non-uniform regions. The OVIS visual display is useful for discovery of these regions and tuning the cluster configuration and environment, as discussed in Section 4.

At this point, if the *physical* environment were uniform, we could just calculate the mean and standard deviation over all nodes and flag as aberrant all nodes whose temperatures fall outside of some probability bounds. A uniform cooling environment, however, is rarely achieved due to physical constraints of air transport. The more typical non-uniform cooling environment presents a challenge to cluster analysis, whether visual or automated, as it is manifest in sometimes non-intuitive ways. This problem is further exacerbated by an environment that can change in unforeseen ways with changes in heat load, supply and return air adjustments, etc. Our solution to this last problem, then, is to create representative environmental models using Bayesian inference. Using these models we can globally normalize node temperature values thus finally allowing us to view them as a large group of statistically similar devices. This type of modeling also allows direct comparison of node temperature with the model as it yields not only a model of the mean distribution but also the associated probability distribution about that mean. Model generation and associated abnormality detection is discussed in Section 5.

4. Visual Abnormality Detection

In order for the GUI display to be most meaningful, either the section of the cluster being viewed must be uniform in factors affecting the quantity of interest (*e.g.*, temperature of CPU 0) or OVIS must normalize the display variable to these factors. For instance, the internal factors affecting the temperature of CPU 0 would be the utilization of CPU 0 and its cooling fan's speed. The result of either method is a display that is ideally uniform other than the statistical variation of values across the cluster. Thus any non-uniformity seen is due to anomalous node behavior or environmental effects. The former is seen as a single node outlier whereas the latter typically has more spatial extent evidenced by a color gradient extending over multiple nodes.

We have used the baseline version of OVIS to examine both raw and statistical behaviors of temperatures in our clusters in order to gain a better understanding of physical cluster configuration effects. We describe two cases in this section involving our commercially obtained Shasta cluster, whose configuration is shown in the OVIS display in Figure 1. The cluster consists of 10 racks depicted as columns in the figure, where blank rows in the center of racks 1-3, and 5 represent empty gaps in the cluster due to networking equipment being mounted on the back side of the racks. Racks 4, 6, and 7, also shown as blank in the figure, contain non-compute equipment.

In the first case, we found that fan speeds for nodes bordering the gaps in the center of racks 1, 2, 3, and 5 were seen running markedly faster than those of their peers as depicted by their fan speed color being much more red shifted. Investigation showed that the basic physical cluster design was flawed as hot exhaust air from the backs of the racks was being recirculated through the gaps into the air intakes of these nodes. Blocking these gaps caused the errant fan speeds to fall in line with the values of the rest of the cluster. The change in fan speed values was immediately apparent by inspection of the OVIS GUI which became uniform in color.

Note that each node taken in singleton was operating well within normal operating parameters. This condition, which could drive these nodes to failure under greater load and/or warmer room conditions, was not even hinted at by the traditional per-node threshold-based monitoring supplied with the cluster.

In another case, we saw that the node temperatures did not increase monotonically with distance from the floor. Rather, the first few nodes nearest the floor had the reverse temperature gradient from what was expected (note the color map in Figure 1, where there are red colored nodes near the floor below green and, hence, cooler nodes). We discovered that the high velocity cooling air from under the floor created a low pressure region at the base of the racks. This, in turn, drew the hot exhaust air from the back of the racks forward, under the racks. In this case, since the fix to the problem is not so simple and the effect was not dangerously large, we chose to leave it be. This type of non-uniform effect, however, makes analysis of the system more difficult as described in the next section.

5. Bayesian Modeling

As mentioned in Section 3, non-uniform environmental effects present a challenge to cluster analysis. In order to account for the effects that we cannot change, we use Bayesian inference to create representative models of these effects. This is implemented in the Bayesian module, which uses these models to identify outliers. Comparison of models is also useful in environmental tuning, as differences in models are reflective of differing environments.

5.1. Environmental Modeling Via Bayesian Inference

It is relatively easy, using the color-coded display, to observe that in our Shasta cluster (Figure 1) CPU temperature varies with height off the floor. How it varies, however, can possibly become very complicated.

As discussed at the end of Section 4, the temperature distribution in the Shasta cluster is non-linear in height, as it is hotter on top and bottom than in the middle. A natural approach thus consists of modeling temperature as some function of height, multiplied by some random "noise" that is on average equal to 1. The noise factor accounts for the parameters other than height that have an effect on temperature; there can be a great many of them, just 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 zero standard deviation would mean an exact fit to the model approximation. We thus fit the observed data to the model

$$T \sim \mathcal{N}(Q(h), \sigma),$$

where h and T denote nodal height and temperature, respectively, $\mathcal{N}(Q(h), \sigma)$ is the normal distribution with mean $Q(h)$ and variance σ, and Q is a polynomial.

To apply this to the Shasta cluster, then, since we know that the dependency of temperature on height is not linear, we need to use a degree that is at least quadratic; on the other hand, visual inspection of the OVIS display of the cluster data values suggests that, qualitatively, the map from height to temperature has only one *minimum* along that curve, and thus that a quadratic polynomial may be used, at least as a first estimate. With this assumption, we are therefore left with 4 unknown parameters in the model: the 3 coefficients of Q, and σ. The method we use to estimate these parameters, based on the data at hand, is Bayesian inference.

The keystone of this approach is Bayes' Theorem which can be informally stated as:

$$P(X|D, M) = P(D|X, M) \times P(X|M)/P(D|M)$$

(or even less formally: posterior = likelihood × prior/evidence) where

- M is the probabilistic model (*e.g.*, temperature for each height is a Gaussian random variable whose mean is a polynomial function of height in the cluster)
- X is a set of model parameters to be inferred (*e.g.*, in the model above, polynomial coefficients of the function and variance of the Gaussian random variable)
- D is the data, *i.e.*, actual values (measurements) of the variables that are present in the model.

Bayesian inference also allows incorporation of expert knowledge in the model; *e.g.*, the fact that temperature T baseline varies with height h has been noted on many systems utilizing under floor air cooling. Another common example of expert knowledge is that of bounds: by simple observation using the OVIS visual display, one can estimate a safe range within which parameters are almost surely contained.

Starting with a given prior that incorporates expert knowledge (such as a uniform prior in the case where only the bounds are known), the Bayesian learning algorithm is iterative, where the posterior becomes the prior each time new data arrives. Since we are interested in the *most probable* parameters to characterize the model, we discard the evidence and only look at the following proportionality (instead of equality) relation:

$$P(X|D, M) \propto P(D|X, M) \times P(X|M).$$

The selected parameter values viewed are those that maximize the posterior probability ("maximum a posteriori (MAP) estimator"). This is a good estimator for a sufficiently peaked probability density distribution. Convergence on a representative model is assumed when a user-defined ratio between two consecutive maximum posterior probabilities is reached, thus yielding the most probable model parameters, conditioned on the data at hand. Then, and conversely, single node comparisons can be made *versus* the inferred model. Since the model describes a normal distribution about a mean described by the inferred polynomial, the 95% probability bounds for the data (conditioned on the parameters previously inferred and the model) are established by a 4σ-wide band and centered about this polynomial. Additionally, the model can be used as a basis for normalization of the nodes it describes.

Though the example models temperature as a function of height, it is by no means restricted to this. This analysis technique can be used to dynamically model any dependencies that can be identified.

5.2. Examples of Bayesian Modeling

5.2.1 Effects of CPU Utilization on models of Temperature vs. Height in our Shasta Cluster

In order to explore how a model may change as a function of computational load, models were generated for our Shasta cluster, whose configuration is shown in Figure 1. We consider 2 cases: 1) Racks 2, 3, and 5 as an aggregate, since they have similar configurations and previous modeling of each of these racks separately yielded very similar models for each, and 2) Rack 8 by itself as its model is very different.

Models were inferred for these cases for each of a series of different CPU utilizations on the nodes (obtained by running OVIS's calibration code which alternates between short idle and calculation periods of adjustable relative duration in order to obtain a relatively stable user-defined CPU utilization on the nodes). In both cases, models were converged upon in about 20-30 timesteps. The resulting polynomials $Q(h)$ are shown in Figure 2, for values of h that correspond to the node heights in the cluster.

With the exception of the idle model in case 1, the resulting polynomials remain basically the same, except for changes in the constant term. This means that we don't have to remodel for every shift in CPU utilization; instead we can use the same model and a calculated offset based on CPU utilization. Since modeling can be computationally intensive and environmental changes are typically slow, this characteristic allows relaxation of the requirement for real-time modeling though not real-time comparison of data with an applicable model. The variation in the idle model in case 1 may be due to averaging effects across the 3 racks and requires further investigation.

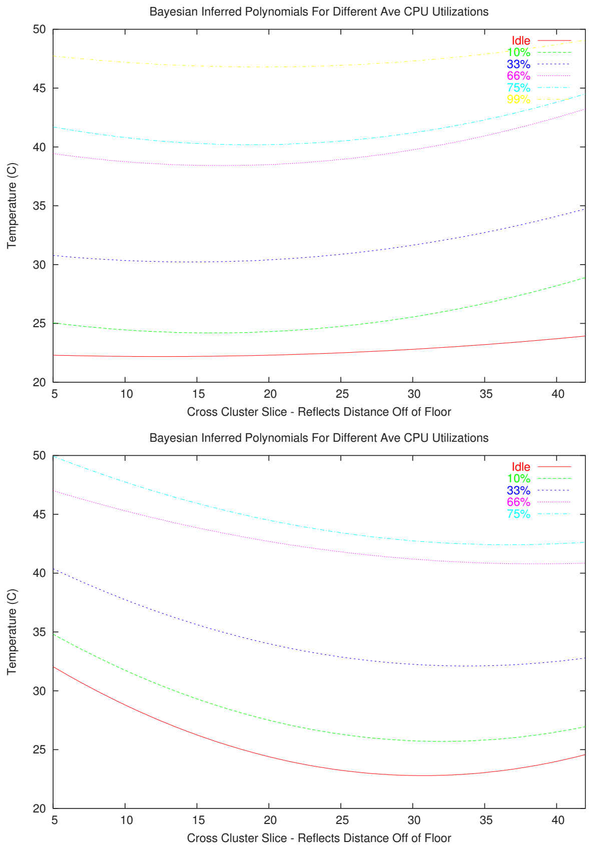


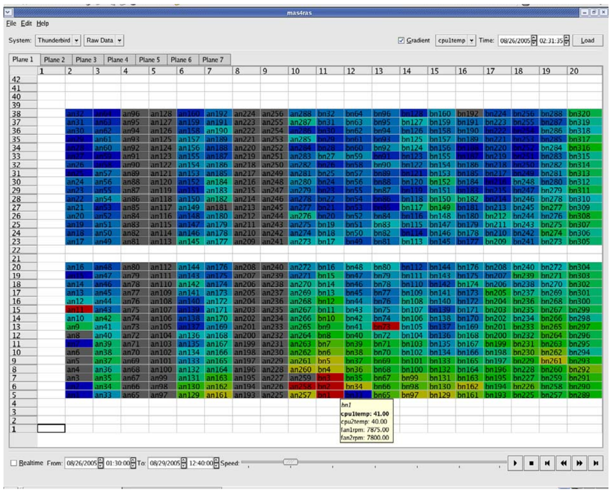
Figure 2. Bayesian inferred polynomials for different CPU utilizations for aggregate Racks 2, 3, and 5 (upper) and Rack 8 (lower).

5.2.2 Dependence of Temperature on Height in the Shasta Cluster

In both cases mentioned in Section 5.2.1 there is variation of temperature with height. Given the effect mentioned previously of high velocity supply air creating a low pressure zone near the bottom of the racks (Section 4), it is immediately apparent from the respective models that this effect is much more pronounced for case 2 than for case 1. Upon inspection we found that a different style of grille with a much higher flow rate was used for racks 8, 9, and 10 and that there was also an opening near the base making the volume of recirculated hot air much greater than for case 1.

5.2.3 Dependence of Temperature on Height in the Thunderbird Cluster

Models were generated for our Thunderbird cluster whose configuration is shown in Figure 3. In particular, we consider racks 11-20 which are the 10 racks on the right side of the figure. These results are for a constant CPU utilization across the cluster.



System: Thunderbird Raw Data

☑ Gradient cpu1temp Time: 08/26/2005 02:31:35

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
42																				
41																				
40																				
39																				
38		an32	an64	an96	an128	an160	an192	an224	an256	an288	bn32	bn64	bn96	bn128	bn160	bn192	bn224	bn256	bn288	bn320
37		an31	an63	an95	an127	an159	an191	an223	an255	an287	bn31	bn63	bn95	bn127	bn159	bn191	bn223	bn255	bn287	bn319
36		an30	an62	an94	an126	an158	an190	an222	an254	an286	bn30	bn62	bn94	bn126	bn158	bn190	bn222	bn254	bn286	bn318
35		an29	an61	an93	an125	an157	an189	an221	an253	an285	bn29	bn61	bn93	bn125	bn157	bn189	bn221	bn253	bn285	bn317
34		an28	an60	an92	an124	an156	an188	an220	an252	an284	bn28	bn60	bn92	bn124	bn156	bn188	bn220	bn252	bn284	bn316
33		an27	an59	an91	an123	an155	an187	an219	an251	an283	bn27	bn59	bn91	bn123	bn155	bn187	bn219	bn251	bn283	bn315
32		an26	an58	an90	an122	an154	an186	an218	an250	an282	bn26	bn58	bn90	bn122	bn154	bn186	bn218	bn250	bn282	bn314
31		an25	an57	an89	an121	an153	an185	an217	an249	an281	bn25	bn57	bn89	bn121	bn153	bn185	bn217	bn249	bn281	bn313
30		an24	an56	an88	an120	an152	an184	an216	an248	an280	bn24	bn56	bn88	bn120	bn152	bn184	bn216	bn248	bn280	bn312
29		an23	an55	an87	an119	an151	an183	an215	an247	an279	bn23	bn55	bn87	bn119	bn151	bn183	bn215	bn247	bn279	bn311
28		an22	an54	an86	an118	an150	an182	an214	an246	an278	bn22	bn54	bn86	bn118	bn150	bn182	bn214	bn246	bn278	bn310
27		an21	an53	an85	an117	an149	an181	an213	an245	an277	bn21	bn53	bn85	bn117	bn149	bn181	bn213	bn245	bn277	bn309
26		an20	an52	an84	an116	an148	an180	an212	an244	an276	bn20	bn52	bn84	bn116	bn148	bn180	bn212	bn244	bn276	bn308
25		an19	an51	an83	an115	an147	an179	an211	an243	an275	bn19	bn51	bn83	bn115	bn147	bn179	bn211	bn243	bn275	bn307
24		an18	an50	an82	an114	an146	an178	an210	an242	an274	bn18	bn50	bn82	bn114	bn146	bn178	bn210	bn242	bn274	bn306
23		an17	an49	an81	an113	an145	an177	an209	an241	an273	bn17	bn49	bn81	bn113	bn145	bn177	bn209	bn241	bn273	bn305
22																				
21																				
20		an16	an48	an80	an112	an144	an176	an208	an240	an272	bn16	bn48	bn80	bn112	bn144	bn176	bn208	bn240	bn272	bn304
19		an15	an47	an79	an111	an143	an175	an207	an239	an271	bn15	bn47	bn79	bn111	bn143	bn175	bn207	bn239	bn271	bn303
18		an14	an46	an78	an110	an142	an174	an206	an238	an270	bn14	bn46	bn78	bn110	bn142	bn174	bn206	bn238	bn270	bn302
17		an13	an45	an77	an109	an141	an173	an205	an237	an269	bn13	bn45	bn77	bn109	bn141	bn173	bn205	bn237	bn269	bn301
16		an12	an44	an76	an108	an140	an172	an204	an236	an268	bn12	bn44	bn76	bn108	bn140	bn172	bn204	bn236	bn268	bn300
15		an11	an43	an75	an107	an139	an171	an203	an235	an267	bn11	bn43	bn75	bn107	bn139	bn171	bn203	bn235	bn267	bn299
14		an10	an42	an74	an106	an138	an170	an202	an234	an266	bn10	bn42	bn74	bn106	bn138	bn170	bn202	bn234	bn266	bn298
13		an9	an41	an73	an105	an137	an169	an201	an233	an265	bn9	bn41	bn73	bn105	bn137	bn169	bn201	bn233	bn265	bn297
12		an8	an40	an72	an104	an136	an168	an200	an232	an264	bn8	bn40	bn72	bn104	bn136	bn168	bn200	bn232	bn264	bn296
11		an7	an39	an71	an103	an135	an167	an199	an231	an263	bn7	bn39	bn71	bn103	bn135	bn167	bn199	bn231	bn263	bn295
10		an6	an38	an70	an102	an134	an166	an198	an230	an262	bn6	bn38	bn70	bn102	bn134	bn166	bn198	bn230	bn262	bn294
9		an5	an37	an69	an101	an133	an165	an197	an229	an261	bn5	bn37	bn69	bn101	bn133	bn165	bn197	bn229	bn261	bn293
8		an4	an36	an68	an100	an132	an164	an196	an228	an260	bn4	bn36	bn68	bn100	bn132	bn164	bn196	bn228	bn260	bn292
7		an3	an35	an67	an99	an131	an163	an195	an227	an259	bn3	bn35	bn67	bn99	bn131	bn163	bn195	bn227	bn259	bn291
6		an2	an34	an66	an98	an130	an162	an194	an226	an258	bn2	bn34	bn66	bn98	bn130	bn162	bn194	bn226	bn258	bn290
5		an1	an33	an65	an97	an129	an161	an193	an225	an257	bn1	bn33	bn65	bn97	bn129	bn161	bn193	bn225	bn257	bn289
4																				
3																				
2																				
1																				

bn1
cpu1temp: 41.00
cpu2temp: 40.00
fan1rpm: 7875.00
fan2rpm: 7800.00

☐ Realtime From: 08/26/2005 01:30:00 To: 08/29/2005 12:40:00 Speed:

Figure 3. OVIS display of a set of racks of our Thunderbird cluster. Racks 11-20 are the 10 racks on the right side of the figure.

The polynomial part of the models, but not the standard deviations, are shown in Figure 4. Due to different environmental conditions, models vary from rack to rack. Racks 13-16 are described by very similar models; Rack 20 has a flatter distribution of temperatures than the other racks.

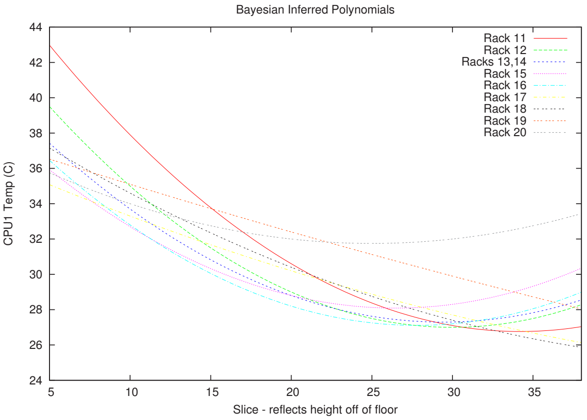


Figure 4. Bayesian inferred polynomials for racks 11-20 of the Thunderbird cluster.

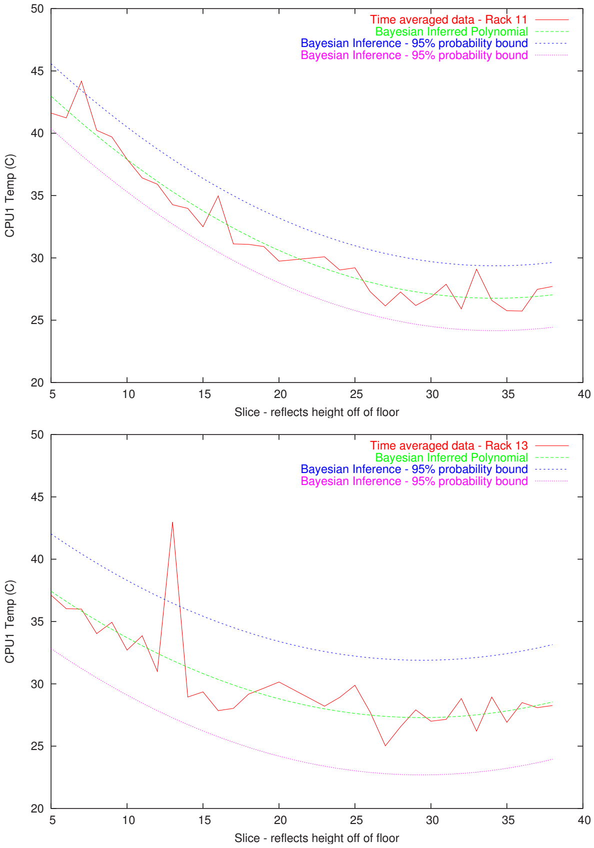


Figure 5. Time averaged data, Bayesian inferred polynomial and 95% probability bounds for racks 11 and 13.

Models for the racks are seen in detail in Figures 5, 6, and 7. In most cases, quadratic polynomial models yield good fits to the data. Rack 15, however, has a more complicated temperature distribution than other racks and, while a 2nd order polynomial fit can be obtained, it is seen empirically not to be a good description of the distribution. In this case the 95% probability bounds allow for a great latitude in the allowable temperatures. Rack 20 is an end rack and has a flatter distribution than that of the internal racks, due to airflow around the end of the cluster.

While rack 13 is in general well described by the quadratic polynomial, it is nonetheless seen to have a value outside the 95% bounds. Automatic detection of abnormalities such as these, is discussed in the next subsection.

5.3. Abnormality Detection

After model inference has been done (with either training or run-time data), we have a stochastic model

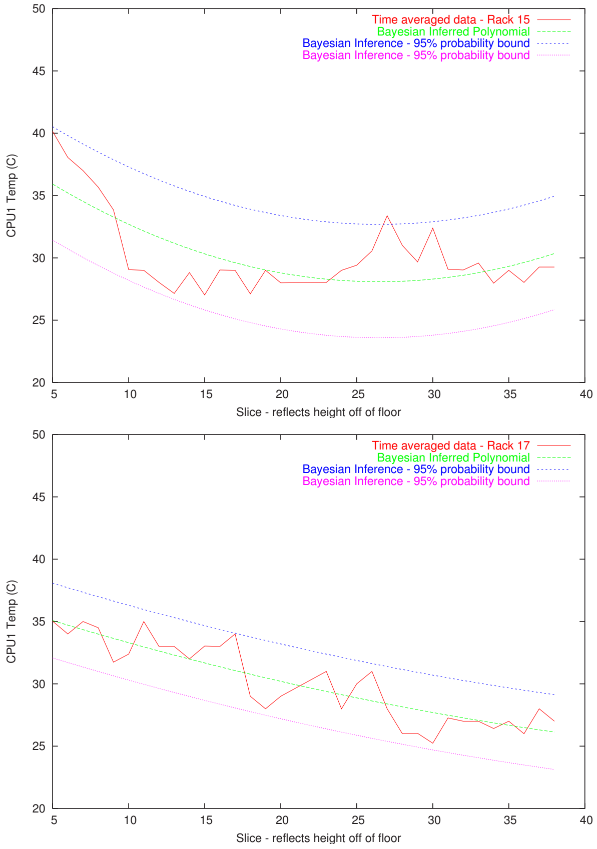


Figure 6. Time averaged data, Bayesian inferred polynomial and 95% probability bounds for racks 15 and 17.

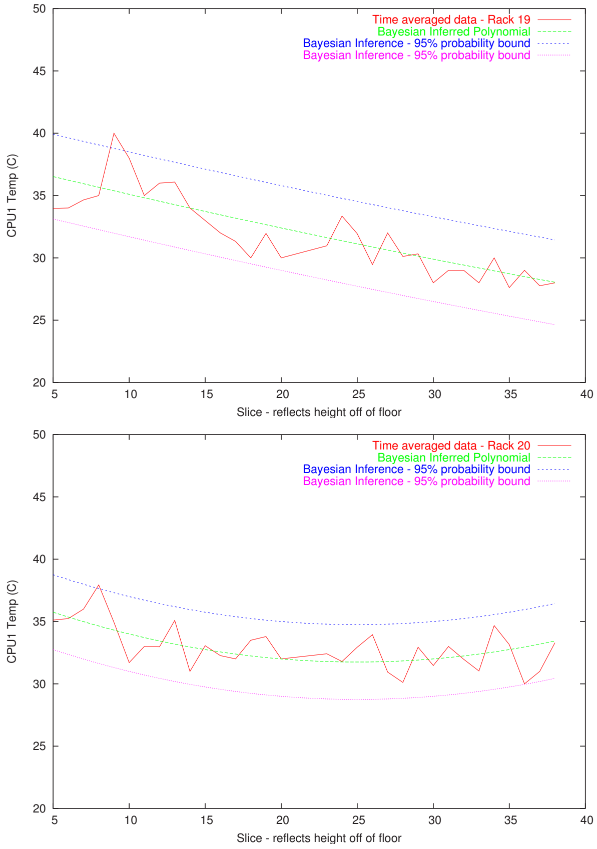


Figure 7. Time averaged data, Bayesian inferred polynomial and 95% probability bounds for racks 19 and 20.

whose parameters optimally fit the data. For example, in a particular rack whose nodes are all idle, the model:

$$T \sim \mathcal{N}(0.005h^2 - 0.1h + 23, 1.5)$$

gives a full (stochastic) description of that rack's node temperature with respect to height. Note that this model is only valid for this point in time and for this particular rack. As new data is acquired, the valid model may change and will certainly be different for a rack with different environmental characteristics.

Such models can be used to determine nodal abnormalities in several ways. One method is to consider whether the data (*e.g.*, run-time data, or stored data for post-crash diagnosis) belongs to a prescribed confidence interval. This can be done either by (1) considering the data relative to applicable models, or by (2) using these models to globally normalize node temperature values and considering these values relative to the resultant global mean (μ) and standard deviation. Then, for example, a 95% confidence interval is established by a 4σ-wide band centered about the mean, $Q(h)$ or μ, respectively. Another useful method is to consider the relative probabilities of the data, given the model. For example, the relative probabilities (probability distribution values relative to the distribution peak) of node at height 10 exhibiting temperatures of 23 or 25, given this model are:

$$RP(h = 10, T = 23|0.005, -0.1, 23, 1.5, M) \approx 95\%$$

$$RP(h = 10, T = 25|0.005, -0.1, 23, 1.5, M) \approx 25\%.$$

Thus, only a few degrees difference in temperature can result in a significant difference in relative probability, depending on the applicable model and what part of the parameter space is being considered. In this paper, we illustrate abnormality detection by comparison of data to their relevant models, as this is the simplest method for automated outlier detection and for clarity in the figures.

Identification of abnormalities based on statistical probabilities allows us to identify potential problems in a cluster much sooner and with greater sensitivity than threshold-crossing mechanisms would. While it is true that “thresholds” are still defined (*e.g.*, the user’s selection of confidence interval), they are now in terms of probabilities determined by statistical processing of actual data rather than a constant defined by the manufacturer. Thus the real numerical threshold values that result are learned and can change in response to aging, environmental effects, etc.

5.3.1 Abnormality Detection in the Shasta Cluster

As in Section 5.2.1, models were inferred for Shasta racks 2, 3, and 5 as an aggregate. The resulting polynomials for average CPU temperature, with 95% probability bounds, both during idle and while hosting a particular application are shown in Figure 8. Individual node values are shown in the figure with bars representing their time-averaged variation.

While other nodes occasionally fall out of bounds, a consistent outlier is seen in slice 10, rack 5, corresponding to node 86. This node runs significantly cooler than other nodes at its height. This effect is much greater under load than at idle.

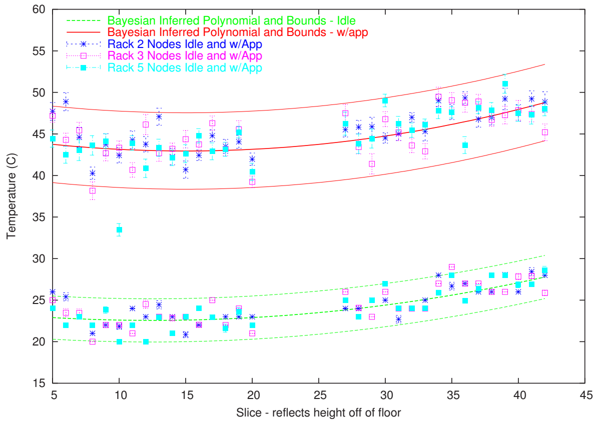


Figure 8. Individual node values and Bayesian inferred polynomials with 95% probability bounds for combined racks 2, 3, and 5 under idle and with-application conditions.

Investigation of this node showed that fan controller failure was causing its fans to run at full speed independent of what the controller was directing/reporting. The fan data did not show this problem, as the values obtained were those the fan controller was reporting, rather than the fan’s actual value.

This is another instance where ensemble consideration is necessary for the discovery of the problem. Traditional thresholding mechanisms consider temperature related failure to occur when a node gets too hot, but not too cold, and therefore low temperature cases are never considered. However, in this case, the low temperature is a manifestation of a problem that could result in reduced lifetime of the fan and possibly, by effect, the node.

6. Conclusions and Future Work

We have presented OVIS, a tool for cluster monitoring and analysis. OVIS’s consideration of nodal values in a statistical rather than a singleton manner enables OVIS to provide more meaningful analysis and detect problems earlier than traditional, threshold-based management tools.

The OVIS graphical display provides deterministic information about state variables and their aggregate statistics. It is particularly useful for detecting environmental effects in the cluster configuration. The Bayesian module enables the system to model out unchangeable environmental effects and to automatically determine abnormalities.

While we limited our discussion in this work to temperature and CPU utilization issues, the methodologies presented here are generally applicable. We will be expanding to additional variables, in particular, memory error rates, voltages, network interface parameters.

Architectural enhancements under development include a distributed framework for the analysis and adaptation of the algorithms for distributed processing. These modifications will also provide fault-tolerance to the design.

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

[5] 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.

[6] R.Vilalta, C. V. Apte, J. L. Hellerstein, S. Ma, and S. M. Weiss. Predictive algorithms in the management of computer systems. *IBM Systems Journal*, 41(3):161–474, 2002.