SAND2015-1074C

A Principled Approach to HPC Event Monitoring

Alireza Goudarzi, Dorian Arnold and Darko Stefanovic
Department of Computer Science
University of New Mexico
alirezag|darnold|darko@cs.unm.edu

Kurt B. Ferreira
Scalable System Software
Sandia National Laboratories[*]
kbferre@sandia.gov

Guy Feldman
Department of Statistics
Purdue University
gfeldman@purdue.edu

ABSTRACT

As high-performance computing (HPC) computing systems become larger and more complex, fault tolerance becomes a greater concern. At the same time, the data volume collected to help in understanding and mitigating hardware and software faults and failures also becomes prohibitively large. We argue that the HPC community must adopt more systematic approaches to system event logging as opposed to the current, ad hoc, strategies based on practitioner intuition and experience. Specifically, we show that event correlation and prediction can increase our understanding of fault behavior and can become critical components of effective fault tolerance strategies. While event correlation and prediction have been used in HPC contexts, we offer new insights about their potential capabilities. Using event logs from the computer failure data repository (cfdr) (1) we use cross and partial correlations to observe conditional correlations in HPC event data; (2) we use information theory to understand the fundamental predictive power of HPC failure data; (3) we study reservoir computing and Bayesian classifiers for failure prediction; and (4) finally, we use principal component analysis to understand to what extent dimensionality reduction can apply to HPC event data. This work results in the following insights that can inform HPC event monitoring: ad hoc correlations or ones based on direct correlations can be deficient or even misleading; highly accurate failure prediction may only require small windows of failure event history; and principal component analysis can significantly reduce HPC event data without loss of relevant information.

1. INTRODUCTION

High-performance computing (HPC) systems, including clusters and supercomputers, have become critical infrastructure for computational science and engineering research and in many commercial enterprises. As such systems become larger and more complex, fault and failure management becomes increasingly important. For tightly-coupled and loosely-coupled resource-intensive applications, unmitigated failures can reduce resource availability and lead to low application performance.

Understanding machine fault and failure behavior is key to the development of effective fault tolerance strategies for HPC systems. Researchers have applied statistical approaches in HPC contexts to help with this understanding. For instance, information theory and event correlation have been used for root cause analyses [10, 11, 9] and to inform failure prediction [12, 3]. Additionally, support vector machines, neural networks and other probabilistic models have been used for failure event predictions [3, 8, 12, 4]. However, none of this research is employed in state-of-the-art system monitoring infrastructures. All aspects of current system monitoring practices employ ad hoc strategies based on practitioner expertise and intuition. Without trivializing the value of system administration experience, we argue that as HPC systems get larger and potential monitor data volumes become prohibitively large, we need a systematic, principled approach to HPC system monitoring: *we should leverage the wide array of current and emerging statistical and data mining techniques to determine both the space (which resources/events) and time (sampling resolution) attributes of HPC resource monitoring systems.*

The fundamental challenge is that complex dependencies amongst HPC system events make event log analysis a nontrivial task. In this work, we explore the usefulness of well-known data analysis techniques (that have been applied successfully in other domains) for HPC system event analysis. Our goal is not (yet) to discover new generalizable insights or insights about specific systems, but to demonstrate that such techniques can be used to obtain such insights in the future.

Accordingly, using event logs from the computer failure data repository (cfdr) [1] in addition to event cross-correlations, here (1) we use partial correlations to observe statistical significance and conditional correlations in HPC event data; (2) motivated by our partial correlation results, we use principal component analysis to understand how much dimensionality reduction can apply to HPC event data; (3) we use information theory to understand the fundamental predictive power

[*] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

of failure event data; and (4) finally, we use a failure event prediction case study to demonstrate the impact of our information theoretic analysis.

We offer the following novel contributions:

- a demonstration that for HPC systems direct correlation used in isolation can suggest false interactions between event time series;
- an approach for reducing the volume of event data that are collected and maintained without reducing the effectiveness of that data;
- a principled approach for understanding the limits of HPC event prediction capabilities; and
- an HPC failure event prediction case study that relies solely on failure event history.

In addition to identifying candidate data analysis methods with high impact potential for HPC event data collection, analysis and prediction, we believe that these contributions may have several broader impacts: (1) they suggest that conditional correlations may be very apparent amongst such data and should be accounted for; (2) dimensionality reduction may be a useful approach to consider for dramatically reducing the impact of event data collection on computation, communication and storage resources; (3) HPC failure prediction approaches may need to consider only small windows of failure event histories and, in fact, considering extraneous data may have negative effects; and (4) probabilistic methods may be better suited for analysis and prediction of HPC events. We believe this is the first work in HPC systems research to show the potential relevance of the statistical significance of apparent correlations and the irrelevance of conditional event correlations as well as the first work to suggest the theoretical predictive potential of HPC failure event data.

The organization of the rest of this paper is as follows: next (Section 2), we describe the data set and the data analysis methods we use in this study. Then, we describe our methodology for applying these methods to our data set, present the results of these applications and discuss the contributions and impacts of these results (Section 3). We then review related works in HPC event analysis (Section 4) after which we conclude with a summary of our main findings and the opportunities they open for future research.

2. BACKGROUND

We survey several well-known statistical techniques including cross and partial correlations, principal component analysis, information theory for temporal information content. In Section 3, we demonstrate the potential these techniques have for HPC resource monitoring.

2.1 Cross and Partial Correlations

Cross and partial correlations are used to analyze direct and conditional correlations between multiple variables. A pairwise cross-correlation characterizes a direct linear correlation between multiple variables at different time lags. Partial correlation is a complementary technique that measures the conditional correlation between two variables controlling for all the other variables. Using the direct and the conditional correlation, one can distinguish direct and indirect interactions between multiple variables in a system.

2.1.1 Cross-correlations

We compute the correlation between two time series X and Y, with means μ_X and μ_Y and standard deviations σ_X and σ_Y, using Pearson's correlation coefficient:

$$\rho(X,Y) = \frac{\langle (X-\mu_X)(Y-\mu_Y)\rangle}{\sigma_X \sigma_Y}, \qquad (1)$$

where $\langle\cdot\rangle$ is the expectation operator. From the equation we see that $-1 \le \rho(X,Y) \le 1$, where a correlation of 1 means perfect correlation; the values of the time series change in the same direction, with the same magnitude. A correlation of -1 means perfect inverse correlation; values of the time series change in opposite direction, with the same magnitude [5, 6].

With lag, τ, between the time series, the calculation is:

$$\rho(X,Y,\tau) = \frac{\langle (X(t)-\mu_X)(Y(t+\tau)-\mu_Y)\rangle}{\sigma_X \sigma_Y}, \qquad (2)$$

This lets us measure the correlation between two events in time. To simplify the notation, we use $\rho_{i,j}(\tau)$ to denote the correlation between variables X_i and X_j at lag τ. The answer to the optimization:

$$\tau^*_{i,j} = \arg\max_{\tau} \rho_{i,j}(\tau) \qquad (3)$$

gives the time lag with the strongest correlation between variables X_i and X_j.

2.1.2 Partial Correlations

Cross-correlations observe the direct correlations between X and Y However, this correlation may only be due to correlation of X and Y with other variables. Partial correlations [5, 6], measure conditional correlations, that is, the correlation between two variables controlling for all the other variables. Partial correlation is useful for root-cause analysis and to filter out spurious statistical correlations.

Using the correlation matrix $C(\tau) = [\rho_{i,j}(\tau)]$, we can calculate partial correlations that measure conditional correlations. Let $\bar{C}(\tau) = [\bar{\rho}_{i,j}(\tau)]$ be the inverse correlation matrix. The partial correlation matrix is calculated as follows: $\bar{C}(\tau) = [\bar{\rho}_{i,j}(\tau)]$, where $\bar{\rho}_{i,j}(\tau)$ is given by:

$$\bar{\rho}_{i,j}(\tau) = \frac{-\rho_{i,j}(\tau)}{\sqrt{\bar{\rho}_{i,i}(\tau)\bar{\rho}_{j,j}(\tau)}}. \qquad (4)$$

2.1.3 Correlation Motifs

Direct and partial correlations can help identify three different structural variable correlation motifs as shown in Figure 1: (a) direct correlation ; (b) partial correlation and (c) both direct and partial correlations.

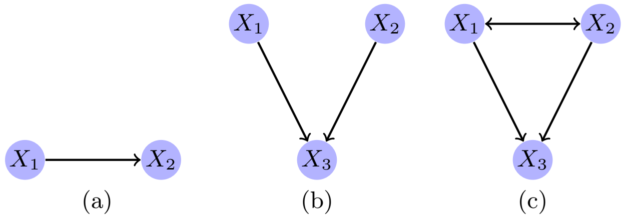


Figure 1: Correlations: (a) X_1 is directly correlated to X_2. (b) X_1 and X_2 are only partially correlated via X_3, (c) X_1 and X_2 are partially correlated via X_3 and also directly correlated.

2.2 Principal Component Analysis

Principal component analysis (PCA) is used to analyze variations in multivariate datasets [2]. PCA transforms a dataset by describing each data point by its variations along its different dimensions. The transformed data are an uncorrelated multivariate datasets; that is, different dimensions are uncorrelated. PCA can be used to sort the data dimensions by total variation and, thus, is used commonly in machine learning for dimensionality reduction in large datasets.

PCA is carried out by calculating the covariance matrix of the data $\mathbf{D}$, its eigenvectors $\mathbf{V} = [\mathbf{v_1}|\ldots|\mathbf{v_n}]$ and its eigenvalues $\lambda = [\lambda_\mathbf{1}, \ldots, \lambda_\mathbf{n}]^\mathbf{T}$. The transformed data can be calculated using $\mathbf{X'} = \mathbf{V} \cdot \mathbf{X^T}$, where X is the original dataset whose columns are different dimensions of the data. The magnitudes of eigenvalues correspond to how much of the total variance is explained by the corresponding column in $\mathbf{X'}$. We calculate the percentage of variation explained by the ith principal component as follows: $100\frac{\lambda_i}{\sum_{j=1}^{n}\lambda_j}$.

2.3 Extracting Temporal Information Content

Information theory provides a framework for measuring information transfer, noise, and loss between an information source and destination. Using information theory, we can identify the predictive information between the past and the future of a time series. In Shannon's information theory [15], entropy is the basic quantity. For an information source S that takes a state $\{s_i | 1 \leq i \leq n\}$ with probability $p(s_i)$, the entropy,$\mathcal{H}_S$, or amount of information in S is defined as:

$$\mathcal{H}_S = -\sum_{i=1}^{n} p(s_i)\log_2 p(s_i). \qquad (5)$$

Joining the entropy of information source and destination:

$$\mathcal{H}_{SD} = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(s_i, d_j)\log_2 p(s_i, d_j). \qquad (6)$$

we can measure how much information is transferred between a source and destination. The mutual information $\mathcal{I}(S : D)$ between a source S and a destination D with states d_j is:

$$\mathcal{I}(S : D) = \mathcal{H}_S + \mathcal{H}_D - \mathcal{H}_{SD}. \qquad (7)$$

We can use mutual information to measure the predictive information in the history of a time series. In this case, the source will be a finite interval in the past of the time series and the destination will be the future value of the time series. Let $y(t)$ be a time-varying signal and let $Y_t, Y_{t-1}, \ldots, Y_{t-(n-1)}$ be the random variables that correspond to the values of y during the last n time steps, i.e., $y(t), y(t-1), \ldots, y(t-(n-1))$. The source S will have the joint probability distribution $p(Y_t, Y_{t-1}, \ldots, Y_{t-(n-1)})$, and the destination D will have the probability distribution $p(Y_{t+1})$.

Information content dictates our (un)certainty about the future of the time series, Y_{t+1}. We can calculate uncertainty U, a measure of our ignorance about the future of a time series, based on our knowledge of its last n values as follows:

$$U = \frac{\mathcal{H}_D - \mathcal{I}(S_t : D)}{\mathcal{H}_D}. \qquad (8)$$

From this definition we observe that $0 \leq U \leq 1$, where $U = 0$ means a complete knowledge of the next step and $U = 1$ indicates a complete ignorance of it.

3. EXPERIMENTS AND RESULTS

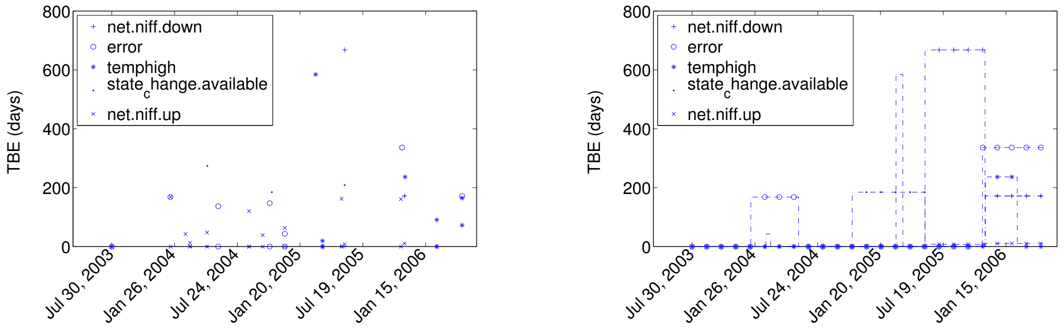
In this work, we argue for a principled, systematic approach HPC resource monitoring and event analyses. We used event and failure data collected from 22 HPC clusters at Los Alamos National Laboratory (LANL) from December 1996 through November 2005 [1]. These systems were large clusters consisting of either non-uniform-memory-access (NUMA) nodes, or 2-way and 4-way symmetric multiprocessing (SMP) nodes. The systems comprised a total of 4750 nodes and 24101 processors. Each failure record included the cluster number and node number for the failed nodes, the failure timestamp, the timestamp at which the node was returned to operational state, and the determined cause of failure. The logs also contained the events from a variety of hardware and system software components. Each event record contained the event timestamp, the node at which the event occurred, the component that produced the event, and the event type. From this repository wee used the more comprehensive "System 20" logs.

3.1 Data Preparation

As described above, our event logs contained information about the node on which the event occurred as well as the component that produced the event, the event category, and the timestamp at which the event was recorded. We needed to preprocess these to convert them into a single multivariate time series in which all events can be analyzed simultaneously. This allowed us potentially to observe the interactions amongst events across different components and different nodes. In the raw event logs, each line denotes a type of event with its corresponding timestamp. Commonly, event correlations are analyzed by calculating either time-between-events (TBEs) or by binning events into sliding time windows along the time series, and generally, TBE calculation and related statistics for a single event is trivial. However, we could not use this approach for two reasons: (1) event frequencies varied greatly among the events and (2) the calculated TBE from the raw data is not synchronized. Therefore, we synchronized the different time series via a *measure-and-hold* schema. That is, for each time series, we set the value of the time series at each time step equal to the calculated TBE at the last occurrence of that event. This synchronized the time series by lengthening all series to span the entire length of the longest time series. Extrapolated time series will not miss any fluctuation in TBE after downsampling. We performed this transformation on all the data and sampled the result at every 7200 seconds (two hours). This improved our analysis efficiency: processing the unsampled data consumed a lot of memory. The result of this extrapolation and synchronization process is shown in Figure 2.

3.2 Finding Event Correlations

We now present the results from our application of the cross and partial correlation methods from Section 2.1. Our HPC cluster event logs contain event data from 2,831 total devices including 512 computer nodes. Analyzing TBEs for all the events logged by all the nodes is a very memory-intensive task. However, since a majority of the nodes logged



(a)

(b)

Figure 2: The time-between-events (TBE) time series before (a) and after (b) extrapolation and synchronization.

only a few events with many months in between, for our event correlation study, we focused on 31 nodes-event pairs that produced the most logged events. We consider both node names and event subcategories to identify each TBE uniquely.

These experiments test correlations amongst events with no time lag. Further, we only consider event pairs with a correlations greater than 0.7 with a p-value < 0.05. (This means that there is only a 5% chance that correlation of the same magnitude can be observed due to random effects.) We highlight our key observations from these experiments:

1. 131 node-event pairs show direct correlation;

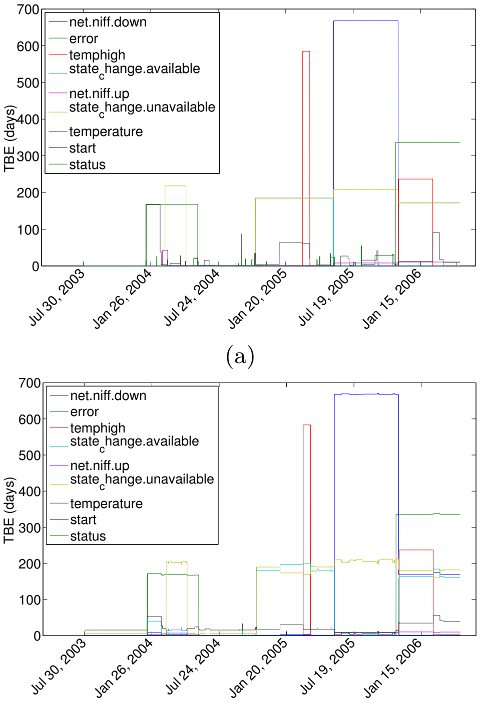
2. 13 node-event pairs show partial correlation; and

3. four node-event pairs show both direct **and** partial correlations.

From the direct correlations in observation 1 alone, one would conclude that there exists a large number of correlated events. However, observation 2 and 3 show that the large majority of these correlations only show up because of the complex interactions amongst a set of event variables and are therefore superfluous. In other words, *there may be a significant difference between the real interactions between event time series and those suggested by direct correlations.*

3.3 Dimensionality Reduction

In Section 3.2 we showed that many correlations are spurious. While these spurious correlations hinder the process of root cause analyses, they suggest that event inter-arrival time series can be compressed significantly and later reconstructed with high fidelity. We now show the results of applying the PCA method for dimensionality reduction (described in Section 2.2) to our data set. Figure 3 show the results of applying PCA to the TBE time series of Node 1 from the data set. Node 1 recorded nine different events in its event logs (Figure 3a). We calculate the principal components of these TBEs and use the five most informative of those to reconstruct the original data (Figure 3b). These five components cover 98.8% of the total variation in the data. This reconstruction is not exact, but it is very close to the original data. In fact, if we normalize the original and reconstructed data between 0 and 1, the average mean-squared-error (MSE) of the nine reconstructed events and the original event is 3.1×10^{-4}.

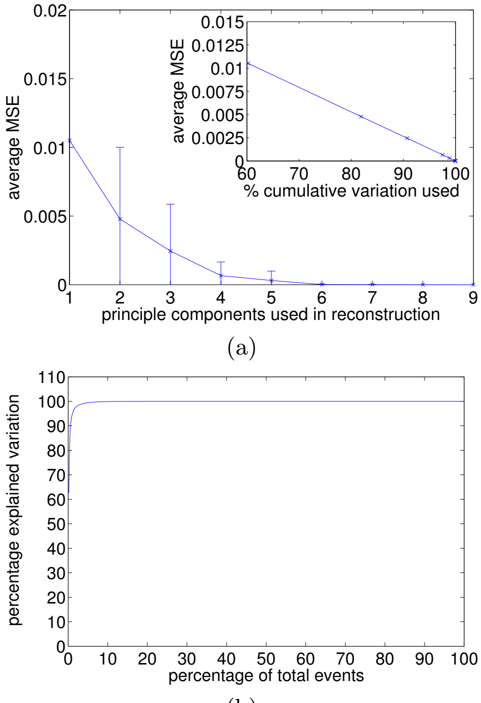


(a)

(b)

Figure 3: The time-between-events (TBE) of nine events logged by Node 1. (a) The original data transformed into continuous multivariate time series as explained in Section 3.1. (b) Reconstruction of the TBE using only 5 most informative components captures a majority of significant events.

Figure 4a shows the average MSE in reconstructing the TBE data as a function of the principal components used. The components are sorted according to the percentage of the variations they explain in descending order. The inset plot shows the MSE as a function of cumulative variation that the principal components explain. The higher the dimensionality of the original data, the fewer components we will need to reconstruct them. To illustrate this we pick the 100 nodes from the data set that record the most events and reconstruct them from principal components. These nodes recorded 574 total events. Figure 4b shows the percent cumulative explained variation as a function of the percentage of the principal components. Only 5% or 29 principal components explain 99.4% of the variation. These results further validate our conclusions from the correlation studies that *event log data can contain a lot of extraneous and unnecessary data.*



(a)

(b)

Figure 4: (a) Average mean-squared-error (MSE) of TBE reconstruction as a function of the number of principal components used. The inset shows the average MSE as a function of percent cumulative variation that the used components explain. (b) Percentage explained variation as a function of the percentage of the most informative principle components. The first 5% of the components explain 94.4% of the variation.

3.4 Temporal Information Content

Generally, event prediction often consists of time-between-event (TBE) prediction. While previous work in event prediction focused on exploiting spatiotemporal correlations among other events, we show that sufficient predictive information about an event's time series is contained within the time series itself. However, mappings between the past and the future TBE values can be highly nonlinear and difficult to discover. We show this using cross-correlations that can determine linear correlations and then use mutual information to find more complex event relationships.

Fu and Xu conducted a comprehensive study of failure correlation and prediction using the moving average of TBE extracted from the same data set that we use [3]. A moving average of a time series can help to reduce measurement errors. We call this the mean k-TBE to indicate that the window length is k steps. Here, we extend this analysis to the information content of the mean k-TBF (time-between-failures) time series and show that, in principle, the history of this time series can be predictive of its future.

First, we construct the mean 8-TBF time series from the data set. We calculate the cross-correlation of the mean 8-TBF and its corresponding p-values. Figure 5 shows the correlations with p-value < 0.05. We observe significant correlations with high values, especially in the first 50 future values of the TBF. However, cross-correlation only signifies the linear correlation between the current and future values of a time series.

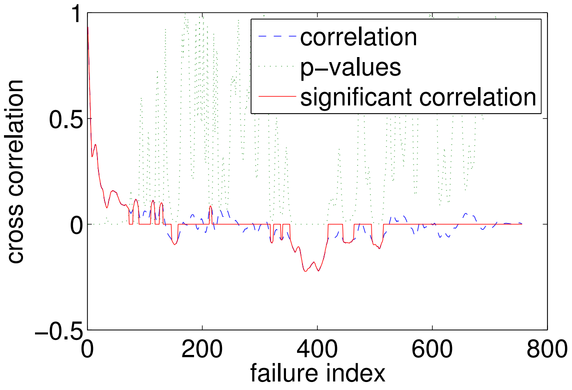


Figure 5: Cross-correlation of the mean 8-TBF. The horizontal axis are TBF index in chronological order. The solid line (red) shows the significant correlation between the TBF and its future. We observe the cross-correlation shows large values within the first 50 indices.

We used mutual information (described in Section 2.3) to study how much information about future time series values is present in the past values of the time series. We first fit an exponential distribution to the original data and use its cumulative distribution function to convert the samples from a uniform distribution. We then discretize the equalized TBF values into 3 bins to avoid overfitting. We calculate the mutual information between a window of length t of the past values of the discretized TBF and its future values. In this case our source was a window of t previous values of TBF S_t and destination will be the next TBF value D.

Recall that our uncertainty measure is a representation that normalizes mutual information from zero to one: as uncertainty approaches zero, the information about the future contained in the previous values of a time series approaches 100%. Figure 6 shows the uncertainty about the next TBF value as a function of the size of the window of past values. As the window size increases, uncertainty rapidly declines, and above window sizes of 15, $U \approx 0.1$. This suggests that *small event histories can contain most of the information about a time series' next step.*

Most of the literature on TBF prediction focuses on predicting the next TBF value. In practical settings, this may not allow sufficient time for remediation strategies to prepare for the failure events or take preventative actions. Therefore, it is most desirable to predict TBFs multiple steps ahead of time. That is, we wish to predict the next several TBF values. For our data set, we calculated the uncertainty for the ith future TBF value knowing the past t values. Figure 7 summarizes the result of this calculation. Once again, we find that the previous 15 steps of the TBF contain most of the information about the next 10 future values of the TBF.

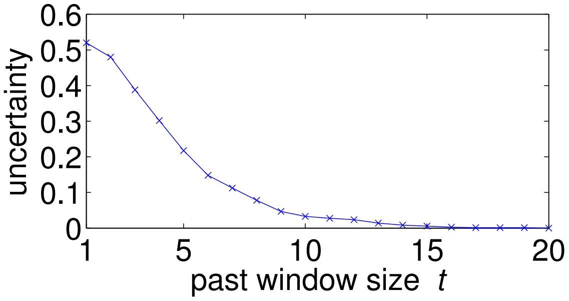


Figure 6: Uncertainty U about the next TBF values based on our knowledge of the past t values of the TBF. With growing window size t, the uncertainty rapidly declines and approaches 0.1 above $t = 15$, which indicates the predictive information in the TBF's history.

In principle, this means that *small event histories can contain the information about a time series' next several steps into the future.*

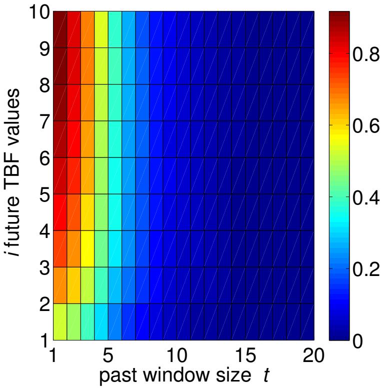


Figure 7: Uncertainty U about the TBF value i steps into the future based on our knowledge of the past t values of the TBF. WIth growing the windows size t, the uncertainty rapidly declines and approaches zero above $t = 15$ which shows that all predictive information about the future of the TBF can be found in its past, in principle.

3.5 Failure Prediction

The results from Section 3.4 suggested to us two things: (1) there is a lot of predictive potential in time series history and (2) the event correlation data might be unnecessary for this purpose. We test these theories by performing a comparison study to that of Fu and Xu [3]. Fu and Xu used a neural network (NN) approach to predict next time series values. Their training set included 450 failure events from September 1, 2003 through August 31, 2004, along with the spatiotemporal correlation between other events in the clusters. They calculated the moving average of the TBF over a window of size $k = 8$ as follows:

$$\hat{m}(t) = \frac{1}{k}\sum_{j=0}^{k} m(t - j + 1), \qquad (9)$$

where $\hat{m}(t)$ and $m(t)$ represent the average TBF and the TBF at time t. We call this 8-TBF. They used a NN with 8 step tapped-delay line and 3 hidden layers with 4 nodes in each layer. They tested their predictor on the 8-TBF time series of the 307 failure events between September 1, 2004 through August 31, 2005. For a sample of length n and prediction values of $\bar{m}(t)$, the prediction error is given by:

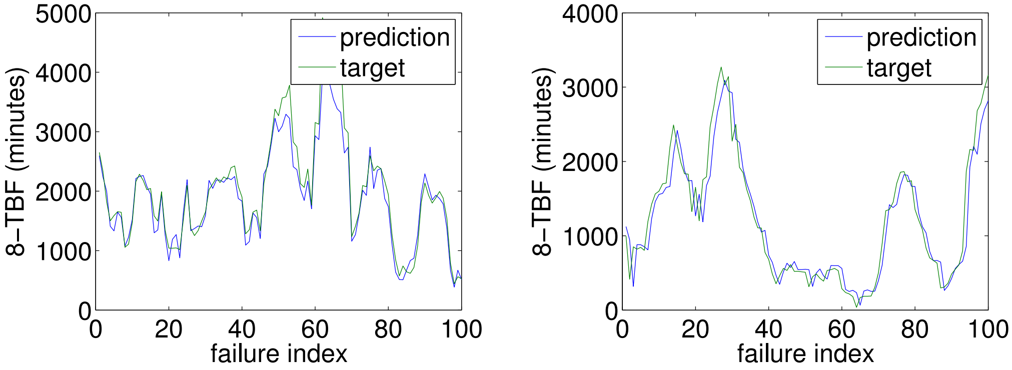
$$\text{error} = \frac{1}{n}\sum_{t=1}^{n} \frac{|\hat{m}(t) - \bar{m}(t)|}{\hat{m}(t)}. \qquad (10)$$

In addition to neural networks, we also investigated the use of echo state networks (ESNs) for time series predictions. We studied ESNs because in other contexts they have demonstrated good success in time series prediction. Before we feed the data to NN and ESN, we normalize the data to values between 0 and 1. The error between the predicted 8-TBF and the actual value is then calculated using the data scaled back to minutes. Figure 8a shows the results when we use a NN with 10 tapped-delay line and a single hidden layer with 12 nodes. Our best observed testing error of 15% is comparable to the 16% reported by Fu and Xu. First, the fact that we are able to perform as well as they did without using the correlation data confirms our finding that information in the TBF time series alone is sufficient to predict its future. Second, we hypothesize that the fact that we actually outperform their results suggests that the extraneous correlation data may perturb negatively the NN's learning and prediction capabilities.

Figure 8b shows the performance of ESN on the prediction task. The error for the best performing ESN is 20%. Since the ESN computation depends on the dynamics of its internal core, the spikes in the time series have a disruptive effect on the performance of the ESN. However, the training in ESN is much faster than standard neural network training, which justifies using ESN in situations where an early approximate value for future TBF is needed.

We can put this result into better perspective. For our predictions, we calculate that the average duration between a failure prediction and its occurrence is 1,697 minutes. Using a result like this, for example as we discuss in the conclusion, failure mitigation strategies can have on average almost three hours to prepare for a component failure. A special concern in TBF prediction is when we predict a larger TBF than the actual value. This is called overestimation and can cause a failure to occur without the proper mitigation in place. The effect of such cases can be easily calculated. For example in our best run, the average overestimation is 175 minutes, which still leaves a lot of time to prepare for the failure.

Next, we performed multi-step prediction in which the predictors had to predict the 8-TBF values i steps into the future. In these experiments, predictors used the 10 previous time steps of the time series; since our results from Section 3.4 suggested more than 10 previous time steps our uncertainty about the future does not increase. We used all 2,478 failure events in the data set and trained the predictors on a window of 300 failure events starting at a random location on the 8-TBF time series. We then tested them on the window of 100 failures immediately following the training interval. We repeated this 10 times and averaged the testing error over the 10 runs. Figure 9 summarizes the results. As we can see from the figure, the prediction error of ESN increases significantly as the future TBF values increase. These increasing errors could be due to the influ-



(a)

(b)

Figure 8: (a) Prediction of the 8-TBF time series for LANL System 20 using neural network. Best observed testing error is 0.15. (b) Prediction of the 8-TBF time series for LANL System 20 using ESN. Best observed testing error is 0.20.

ences of dynamical effects of the time series on the internal dynamics of the ESN.

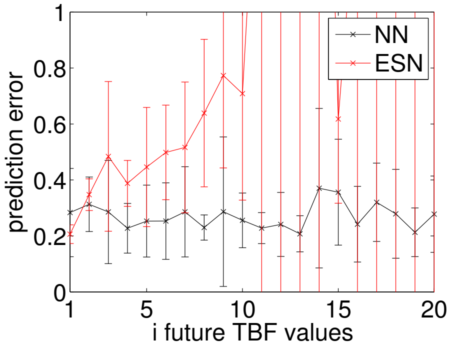


Figure 9: Multi-step prediction result of 8-TBF using neural network (NN) and echo state network (ESN). However, the prediction error of ESN increases significantly, which maybe due to the dynamical effects of the time series on the internal dynamics of the ESN.

A legitimate concern with the use of predictive methods in HPC systems based on historical data is that the system's behavior is not stationary, that is, the statistical properties of the system change over time. This is usually reflected in the changing average and variance of the signals extracted from the system. In Section 3.4 we showed that uncertainty U about the future of the TBF is a function of a small window of its previous values. These results are calculated over the entire TBF time series that contain 2,478 values. However, we can calculate U over a limited time interval and slide this interval over the entire time series to see how U changes. To study this, we calculate U over a sliding interval of 500 consecutive 8-TBF time series. Figure 10a shows the evolution of U over time characterized by periods of stability and minor fluctuations around a well-defined value interleaved by rapid jumps to another stable regime. Figure 10b shows the distribution of uncertainty value over the entire failure time series. This distribution is well-behaved and centered around a well-defined value. This suggest that although the local behavior of the system may show rapid fluctuations, the global behavior is more or less well behaved. This justifies that predictive approaches based on historical value may be useful. However, a second implication is that near the rapid jumps the prediction performance may suffer.

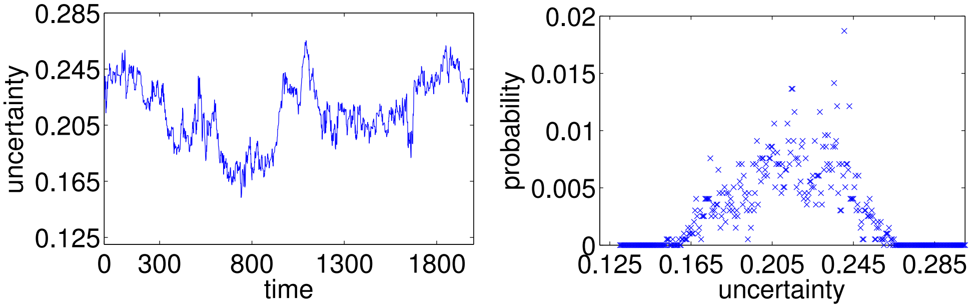
4. RELATED WORK

Several studies have shown important spatial and temporal correlations amongst failure events. Schroeder and Gibson studied failure event logs from many high performance clusters and showed important temporal correlations between failure events [14]. The day-of-week and hour-of-day correlation patterns in this work are useful for scheduled tasks on HPC systems, but is not suitable in informing adaptive and predictive approaches of fault management. Based on failure data from BlueGene/L at the IBM Thomas J. Watson Research Center, researchers used log filtering techniques to calculate the alert inter-arrival time. They showed significant correlations between failure events, both in space and time [13, 7]. Finally, Oliner, Kulkarni, and Aiken discovered a structure of causal influence between the components of HPC systems [11], by calculating time-lagged cross-correlation and analyzing how the signals from different system components deviate from their normal behavior.

While event cross-correlations have been used in HPC system studies, partial correlations have not been well studied. Using both cross-correlation and partial correlation can provide additional insight into the complex interactions between multiple components that are hard if not impossible to obtain using cross-correlation alone. This includes the statistical significance of correlations as well as deciding whether events are directly or only indirectly related.

5. CONCLUSIONS

In this study, we demonstrated the discrepancy between the real interactions between event time series and those suggested by direct correlations. Furthermore, we show that principal component analysis can be used effectively for dimensionality reduction of HPC data. Moreover, we demonstrated a principled approach for understanding the limits of HPC event predictability and showed that accurate failure event prediction can rely solely on failure history, and no further event information.



(a) Evolution of uncertainty U over time

(b) Probability distribution of U

Figure 10: Evolution of the uncertainty U over time and its corresponding distribution. The centered distribution indicates that the function is well behaved and justifies the prediction based historical values.

We believe this work can and should influence resource monitoring practice for current and future HPC systems. Event correlation techniques, such as partial correlation, that filter out noisy conditional correlations can lead to more accurate and more efficient failure diagnoses, failure dependence analyses and root cause analyses. Methods such as partial correlation and PCA can reduce event data volumes without loss of information or predictive effectiveness. This allows us to consider optimizations to HPC event data collection and management. For example, we can reduce data volume via event filtering or dimensionality reduction. Or we can take more drastic measures, such as only monitoring and collecting events we **know** to be directly relevant for other events that we care about. Finally, understanding the theoretical limits of the predictive power of data sets lets us know how close our prediction tools are to their peak capabilities, so that we know when we have approached the point of diminishing returns. Also, knowing what data is relevant for effective predictions and monitoring and collecting only that minimal set renders simpler, more efficient predictors (such as neural networks) that improve prediction time and, as our results have shown, prediction accuracy.

6. REFERENCES

[1] The computer failure data repository (CFDR). "`https://www.usenix.org/cfdr`", Online; accessed August 3, 2013.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] Song Fu and Cheng-Zhong Xu. Exploring event correlation for failure prediction in coalitions of clusters. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, SC '07, pages 41:1–41:12, New York, NY, USA, 2007. ACM.

[4] Errin W. Fulp, Glenn A. Fink, and Jereme N. Haack. Predicting computer system failures using support vector machines. In *Proceedings of the First USENIX conference on Analysis of system logs*, WASL'08, pages 5–5, Berkeley, CA, USA, 2008. USENIX Association.

[5] J.D. Gibbons. *Nonparametric Statistical Inference*. Marcel Dekker Inc, 2nd edition, 1985.

[6] M.G. Kendall. *Rank Correlation Methods*. Griffin, 1970.

[7] Y. Liang, Y. Zhang, Anand Sivasubramaniam, R.K. Sahoo, J. Moreira, and M. Gupta. Filtering failure logs for a bluegene/l prototype. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pages 476–485, 2005.

[8] Yinglung Liang, Yanyong Zhang, Anand Sivasubramaniam, Morris Jette, and Ramendra Sahoo. Bluegene/l failure analysis and prediction models. In *Proceedings of the International Conference on Dependable Systems and Networks*, DSN '06, pages 425–434, Washington, DC, USA, 2006. IEEE Computer Society.

[9] A. Oliner and J. Stearley. What supercomputers say: A study of five system logs. In *Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on*, pages 575–584, 2007.

[10] A.J. Oliner, A. Aiken, and J. Stearley. Alert detection in system logs. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 959–964, 2008.

[11] A.J. Oliner, A.V. Kulkarni, and A. Aiken. Using correlated surprise to infer shared influence. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 191–200, 2010.

[12] R. K. Sahoo, A. J. Oliner, I. Rish, M. Gupta, J. E. Moreira, S. Ma, R. Vilalta, and A. Sivasubramaniam. Critical event prediction for proactive management in large-scale computer clusters. In *In Proceedings of the 9th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining*, pages 426–435. ACM Press, 2003.

[13] R.K. Sahoo, M.S. Squillante, A. Sivasubramaniam, and Y. Zhang. Failure data analysis of a large-scale heterogeneous server environment. In *Dependable Systems and Networks, 2004 International Conference on*, pages 772–781, 2004.

[14] B. Schroeder and G.A. Gibson. A large-scale study of failures in high-performance computing systems. In *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*, pages 249–258, 2006.

[15] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 1948.