

Reliability Analysis in HPC clusters

Narasimha Raju, Gottumukkala, Yudan Liu, Chokchai Box Leangsuksun¹,

Raja Nassar, Stephen Scott²

College of Engineering & Science, Louisiana Tech University

Oak Ridge National Lab²

{nrg003,yli010, box, nassar}@latech.edu, sscott@ornl.gov

Abstract

Resource failures and down times have become a growing concern for large-scale computational platforms, as they tend to have an adverse affect on the performance of the computation system. Reliability-aware resource allocation and checkpointing algorithms have been investigated to minimize the performance loss due to unexpected failures. The effectiveness of a reliability-aware policy relies on the accuracy of reliability prediction. The reliability of a group of nodes is evaluated as a combination of individual node information under an assumption that each node reliability is independent. In this paper, we describe the reliability analysis based on time between failures for a system/group of nodes. Various reliability models are compared for different cases in order to find an optimal reliability model. The reliability models and analysis techniques were evaluated based on actual failure data of 512 nodes during their four year operational period.¹

1. Introduction

Increasing demand for computing power in scientific and engineering applications has spurred deployment of (HPC) high performance computing systems that deliver tera-scale performance. Current and future HPC systems that are capable of running large-scale parallel applications, may span hundreds of thousands of nodes. In fact, top500.org reports the current highest processor count to be 131K nodes [9]. For parallel programs, the failure probability increases significantly with the increase in number of nodes. Ignoring failures or system reliability can have severe effect on the performance of the HPC cluster, and quality of service. A reliability monitoring and analysis framework [4] provides up-to-date reliability of selected components. A resource

manager can use the reliability information of nodes to schedule a parallel job on a set of nodes to minimize job completion time. A checkpoint algorithm can also use the information that may enable to schedule a checkpoint based on the failure probability of selected nodes [10] in order to minimize the performance loss.

The reliability of a set of nodes constitutes the individual node failure information. System reliability is the reliability of one or more nodes and/or components that are required for successful running of a parallel application. In this paper, we analyze the system reliability by considering the failure events of the system comprising k nodes. Based on the reliability monitoring and analysis framework from our prior work [4] we present an approach to calculate the system reliability of k nodes from failure event logs of individual nodes. We also compare various reliability models based on our approach of calculating system reliability.

The rest of the paper is organized as follows. Section 2 discusses the related work in reliability modeling and failure prediction for HPC. Section 3 briefly describes various categories of failure events. Section 4 presents the time to failure reliability models, and proposed algorithm for combining time to failure data of k nodes. Section 5 entails the results of comparing reliability models when different numbers of nodes are selected, and finally section 6 presents the conclusion and future work.

2. Related Work

Failures in large-scale HPC systems have adverse impact both on the performance and quality of service. There have been efforts to predict failures based on historical data, and failure-aware scheduling algorithms [1]

[7][8]. Failure prediction based on individual events

¹Research supported by the Department of Energy Grant no: DE-FG02-05ER25659.

²Research supported by the Mathematics, Information and Computational Sciences Office, Office of Advanced Scientific Computing Research, Office of Science, U. S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

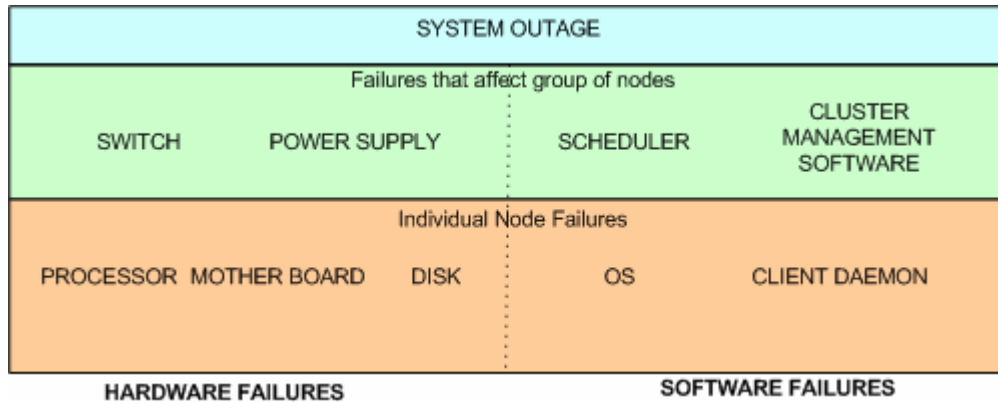


Figure 1: Description of Failures in a HPC platform

is still an ongoing research, and may not be possible to predict the exact type and time of failure. While current failure prediction techniques are not suitable to be applied as an alternative to enable fault tolerance mechanism, the failure probability can be used to develop reliability-aware schemes [10][8][5] like checkpointing to minimize performance loss. Our reliability analysis is based on widely used statistical reliability models.

3. Failures in large scale HPC systems

For a parallel program running on a set of k nodes, if any component in a node, or a component common to the set of nodes (e.g. network hub) fails, the program [6], a 512 Node cluster from LLNL. Each node is a 16 SMP (Symmetric multi processor), and so there are totally 8192 processors. The failures and down times of each Node were collected over a 4 year 3 month time period from 7/15/2000 to 10/1/2004, (total of 37218 hours). The raw failure log consists of the date and time of failure event, down time, type and a very brief description of failures. Some of the failures, which we think do not affect the job, were filtered out. For example, some failures were only repaired after a long time, and did not affect the job runtime.

4. Time to Failure distribution

The failure data consists of failure times, and down times of the nodes. The time between failures is assumed to be a random variable which follows a certain distribution. The actual failure times are obtained by subtracting the down times (see Figure 1). That is, suppose a node fails at times, f_1, f_2, f_3 , we fit the distribution of time between failures (intervals) i.e f_2-f_1 ,

running on all the k nodes is affected. Figure 1 shows examples of hardware and software failures that affect a single node, a group of nodes and all the nodes. In the failure trace, the failure events that affect more than one node are recorded as failure events for those nodes that were impacted. When failure events of each of k nodes are combined, common failures are treated as one to avoid duplication when analyzing the system of nodes.

Failure Data

For our analysis, we use failure data of ASC White machine

f_3-f_2, f_4-f_3 , and obtain $F(t)$ the cumulative failure times for individual nodes. We obtain individual node reliabilities R_1, R_2, \dots, R_N from $R = 1 - F(t)$

4.1 Time between Failures for a system of k nodes

Parallel applications are allocated a set of k nodes for execution. Each node has an individual failure distribution. In our model, the system fails, when at least one node fails. Thus, we are interested in the minimum time till failure when k nodes are combined. The time till failure of individual nodes are combined to obtain the time to failure distribution of k nodes when the first failure occurs. The algorithm for combining time between failures from individual nodes is described in figure 4. We compare three different distributions in our reliability models namely Exponential, Weibull and Lognormal. Figure 4 shows the CDF's (Cumulative Distribution Function) of the reliability models.

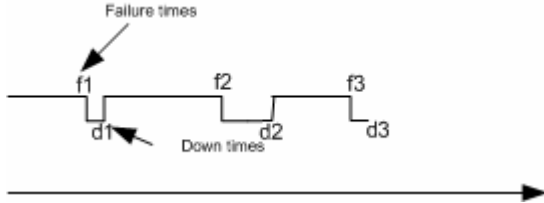


Figure 2 Time between failures obtained from failure logs by removing downtimes, and calculating the difference between times.

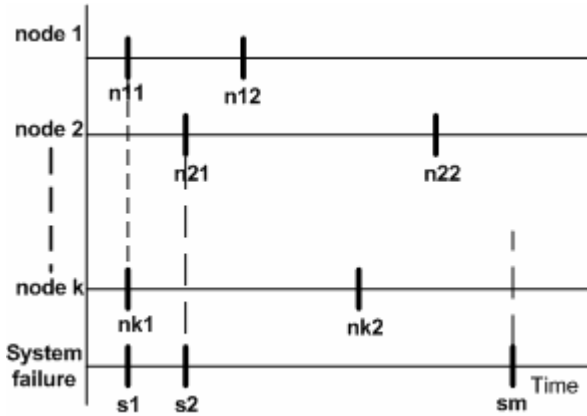


Figure 3. Time between failures of a system of k nodes

Algorithm for calculating the time between failure for a system of k nodes

$T_{n1}, T_{n2}, \dots, T_{nk}$ are the set of failure times on node 1, node 2, ..., node k

$$T_{n1} = \{n_{11}, n_{12}, n_{13}, \dots\}$$

$$T_{n2} = \{n_{21}, n_{22}, n_{23}, \dots\}$$

.....

$$T_{nk} = \{n_{k1}, n_{k2}, n_{k3}, \dots\}$$

where n_{k2} means the start time of second failure on node k

For a system of k nodes, the failure time set F_s is

$$T_s = T_{n1} \cup T_{n2} \cup \dots \cup T_{nk} = \bigcup_{i=1}^k T_{ni} = \{s_1, s_2, s_3, \dots, s_m\}$$

The time between failure (TBF) of the k nodes system is:

$$TBF_s = \{s_{i+1} - s_i\}, i = 1, 2, 3, \dots, m$$

5. Comparison Results

We calculate the reliability of 'k' nodes by analyzing the time between failures of the system comprising the k nodes. The time between failures is calculated using the algorithm given in section 4. Different failure distribution for exponential, weibull and lognormal are then compared with empirical failure distribution. We assume that parallel programs are usually allocated to 2^n processors. We show the time to failure distributions for $k = 4, 64, 128$ and 256 . We compare the goodness of fit when nodes are selected randomly, and when nodes are selected in order in Table 1.

Kolmogorov- Smirnov (K-S) goodness of fit test is used to compare the distributions. The K-S goodness of fit test gives the maximum distance between the empirical and theoretical distribution. A p-value equal or less than 0.05 indicates that the distribution does not fit the data. Table 2 shows the goodness of fit for two cases, when nodes are selected randomly, and when nodes are selected according to node number. In both cases of our experiment, weibull is observed to be a better model for reliability of a system of k nodes as compared to exponential and lognormal fit.

| Distribution | CDF |
|--------------|--|
| Exponential | $F(t) = 1 - e^{-\lambda t}$ λ - e failure rate |
| Weibull | $F(t) = 1 - e^{-(t/c)^m}$ c - characteristic life, m - shape parameter |

| Distribution | CDF |
|--------------|--|
| Log Normal | $F(t) = \Phi \left\{ \frac{\ln \frac{t}{T_{50}}}{\sigma} \right\}$ σ - shape parameter T_{50} - medial life at 50% failure point |

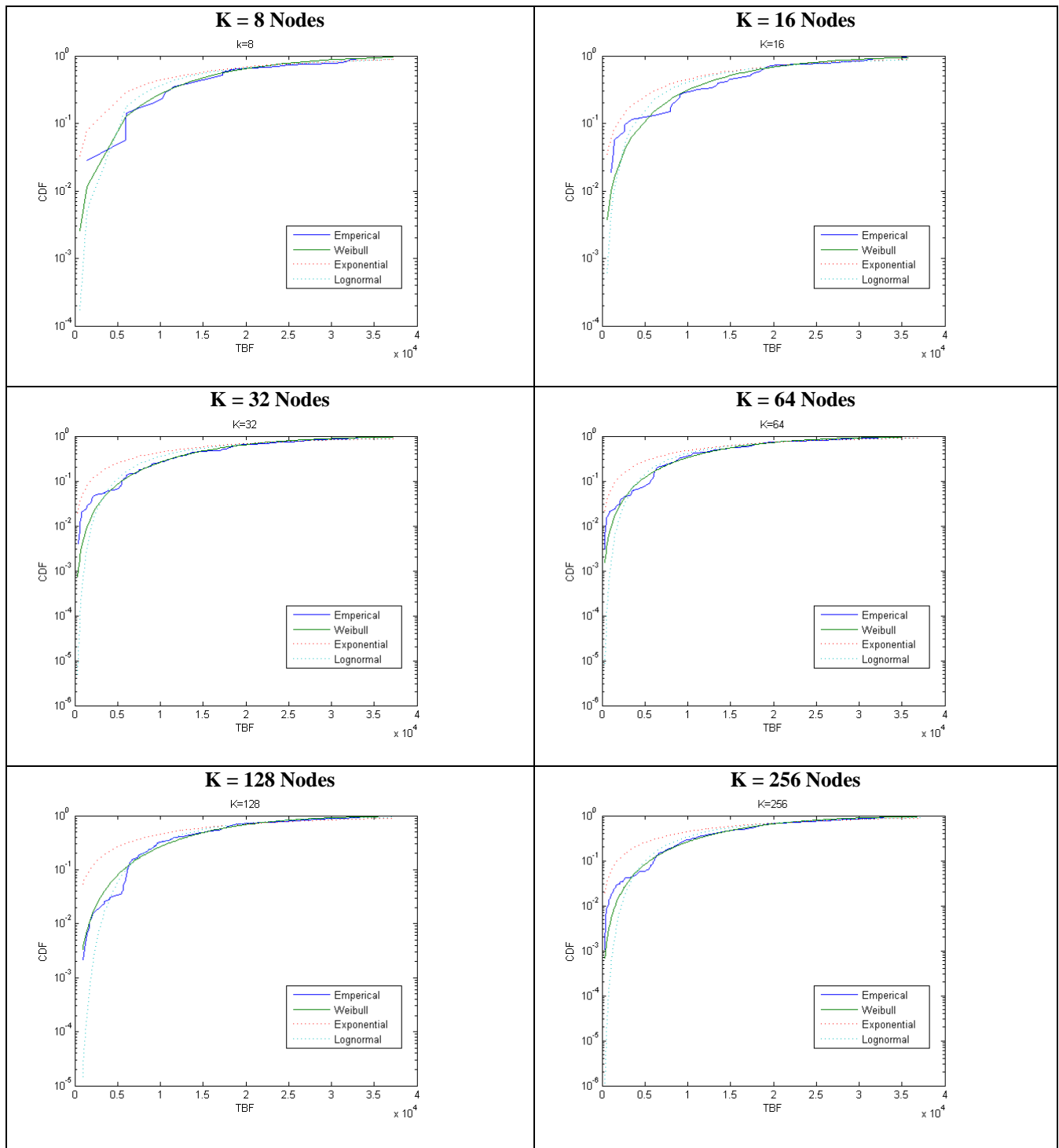


Figure 4. Comparison of empirical and Weibull CDFs when k=4, 64,128 and 256 nodes are selected.

Table 1: Comparison of Failure distributions using the Kolmogorov-Smirnov Goodness of Fit Test

| Comparison of Kolmogorov-Smirnov Goodness-of-Fit Test of various number of nodes (selected in order) | | | | Comparison of Kolmogorov-Smirnov Goodness-of-Fit Test of various number of nodes (selected Randomly) | | | |
|--|-------------|---------------|---------------|--|---------------------|-----------------|-------------------|
| No of Nodes | p-value | p-value | p-value | No of Nodes | Exponential p-value | Weibull p-value | Lognormal p-value |
| K | Exponential | Weibull | Lognormal | | | | |
| 2 | 0.2628 | 0.8679 | 0.5409 | 2 | 0.6060 | 0.4460 | 0.6573 |
| 4 | 0.2049 | 0.4310 | 0.9034 | 4 | 0.9940 | 0.8151 | 0.9852 |
| 8 | 0.1916 | 0.9980 | 0.8571 | 8 | 0.2272 | 0.5758 | 0.7485 |
| 16 | 0.0818 | 0.9845 | 0.3269 | 16 | 0.3193 | 0.7091 | 0.4671 |
| 32 | 0.0002 | 0.6300 | 0.0438 | 32 | 0.4829 | 0.4829 | 0.2460 |
| 64 | 0.0000 | 0.7122 | 0.1538 | 64 | 0.0224 | 0.2484 | 0.0785 |
| 128 | 0.0000 | 0.2652 | 0.0779 | 128 | 0.0000 | 0.1169 | 0.0061 |
| 256 | 0.0000 | 0.0599 | 0.0000 | 256 | 0.0000 | 0.0453 | 0.0000 |
| 350 | 0.0000 | 0.0388 | 0.0000 | 350 | 0.0000 | 0.0159 | 0.0000 |

6. Conclusion and Future work

Failures and downtimes are a growing concern for large scale HPC systems. The system performance and quality of service can be adversely affected due to resource outages. Current checkpoint based fault tolerance can have a significant overhead. In addition, failure prediction based on prior events is still in initial stages of research. Thus, we believe that reliability-aware approach such as a resource manager can exploit the reliability information to better allocate a particular job so that the job completion time is minimized. In addition, the reliability-aware checkpointing algorithms can schedule a checkpoint at intervals based on reliability of resources to minimize checkpoint overhead. These services require reliability information when a parallel application is allocated to a singled node or a group of nodes.

In this paper, we described an approach to evaluate the reliability of a single node, and then a system of k nodes. Reliability is estimated based on the time between failures data obtained from the failure history of nodes. Using the actual failure trace obtained from prominent HPC platform, we studied and compared appropriateness of different distributions, Exponential, Weibull and Lognormal for various cases systems of k nodes. Our results indicate that Weibull distribution results in the better reliability model in most of the cases for the given data. In the future, we plan to investigate the performance improvement in resource management and checkpoint interval selection with different reliability prediction techniques.

7. References:

- [1] A. J. Oliner, R. Sahoo, J. E. Moreira, M. Gupta, and A. Sivasubramaniam. Fault-aware Job Scheduling for BlueGene/L Systems. In Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), 2004.
- [2] Engelmann and G. A. Geist. "Super-Scalable Algorithms for Computing on 100,00 Processors". Proceedings of International Conference on Computational Science (ICCS), Atlanta, GA, USA, May 2005.
- [3] Elmootazbellah N. Elnozahy, James S. Plank. "Checkpointing for Peta-Scale Systems: A Look into the Future of Practical Rollback-Recovery," IEEE Transactions on Dependable and Secure Computing, vol. 01, no. 2, pp. 97-108, April-June, 2004.
- [4] H. Song, C. Leangsuksun, N. R. Gottumukkala, S. L. Scott, and A. Yoo. Near-real-time availability monitoring and modeling for HPC/HEC runtime system. In Proceedings of Los Alamos Computer Science Institute (LACSI) Symposium 2005, Santa Fe, NM, USA, October 11-13, 2005.
- [5] James S. Plank and Michael G. Thomason, "The Average Availability of Parallel Checkpointing Systems and Its Importance in Selecting Runtime Parameters," 29th International Symposium on Fault-Tolerant Computing, Madison, WI, June, 1999, pp. 250-259.

- [6] Lawrence Livermore National Laboratory
Trace Logs:
[url:http://www.llnl.gov/asci/platforms/white/](http://www.llnl.gov/asci/platforms/white/)
- [7] 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 Proceedings of the ACM SIGKDD, Intl. Conf. on Knowledge Discovery Data Mining, pages 426–435, August 2003.
- [8] Schroeder, B. and Gibson, G. A. 2006. A large-scale study of failures in high-performance computing systems. In Proceedings of the international Conference on Dependable Systems and Networks, June 2006.
- [9] Top 500 Super Computing Sites List, July 2006, : url: <http://www.top500.org/>
- [10] Y. Liu and C. B. Leangsuksun. "Reliability-aware Checkpoint/Restart Scheme: A Performability Trade-off". Submitted to IEEE Cluster Computing (Cluster), Boston, MA, USA, September 2005.
- [11] Yinglung Liang, Yanyong Zhang, Anand Sivasubramaniam, Morris Jette, Ramendra Sahoo, "BlueGene/L Failure Analysis and Prediction Models," dsn, pp. 425-434, International Conference on Dependable Systems and Networks (DSN'06), 2006.