Doomsday

Anwesha Das, Frank Mueller, Paul Hargrove, Eric Roman, Scott Baden Lawrence

Introduction

HPC systems are expensive computing environments composed of hundreds or thousands of nodes with non-uniform memory access.

Like everything distributed, individual nodes can fail. Because we want high performance, failure is *very* expensive.

We can reduce the overhead of failure recovery if we can predict the failures proactively in these large scale computing systems.

Motivation

Existing work does not place sufficient emphasis on lead time requirements.

Prior studies use the same training data for future predictions over a long time frame. Dynamic prediction and scalable online prediction techniques have not yet been explored.

Most studies have focussed on rich BlueGene logs of decommissioned systems. Contemporary systems(e.g. Cray) with lower-level Linux style raw logs need further exploration

Proposal

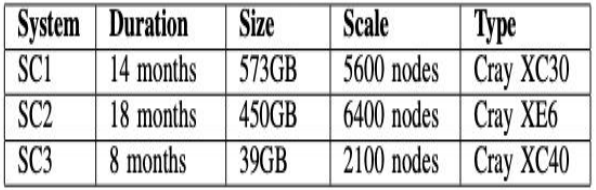
The paper proposes a novel prediction scheme,TBP(time based phrase) to extract relevant log phrases indicative of node failure from noisy data.

These events help forecast future failures with lead times ranging from 20 secs to 2 minutes.

Cray System Architecture

Scale : These systems have been widely deployed and typically run more than 1,400,000 jobs/year.

TABLE I: System Details



System	Duration	Size	Scale	Type
SC1	14 months	573GB	5600 nodes	Cray XC30
SC2	18 months	450GB	6400 nodes	Cray XE6
SC3	8 months	39GB	2100 nodes	Cray XC40

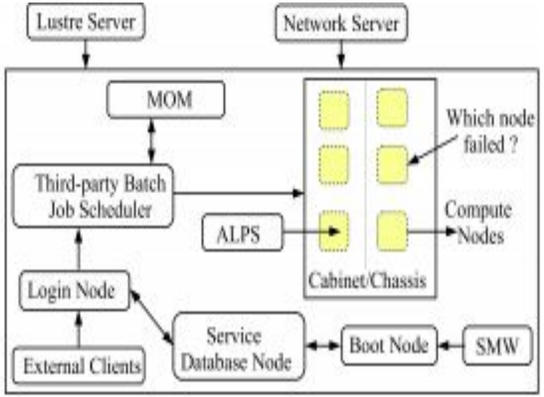


Fig. 1: Overview of a Cray System

Technical Challenges

Failure needs to be discovered by integrating a distributed set of events over space and time.

Normalizing, Mapping, Asymmetric Binarization of data cannot reveal the information required.

Non-critical messages could be better predictors.

Errors propagate in the system making it harder to find a correlation between distant error logs.

What is Node Failure?

Broadly speaking, node failures can be classified as Internal Failures, External Failures, Normal Shutdowns.

Normal Shutdowns are administrative events like maintenance.

Internal Failures are specific to the node at hand and are not influenced by the state of the system.

External Failures are triggered by errors or failures in other parts of the system.

Example

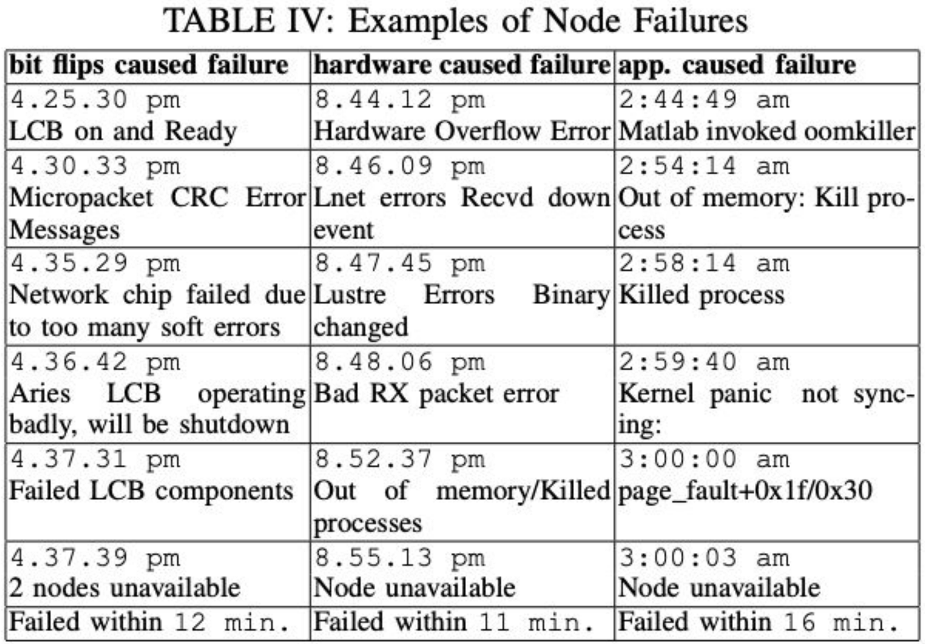


TABLE IV: Examples of Node Failures

bit flips caused failure	**hardware caused failure**	**app. caused failure**
4.25.30 pm LCB on and Ready	8.44.12 pm Hardware Overflow Error	2:44:49 am Matlab invoked oomkiller
4.30.33 pm Micropacket CRC Error Messages	8.46.09 pm Lnet errors Recvd down event	2:54:14 am Out of memory: Kill process
4.35.29 pm Network chip failed due to too many soft errors	8.47.45 pm Lustre Errors Binary changed	2:58:14 am Killed process
4.36.42 pm Aries LCB operating badly, will be shutdown	8.48.06 pm Bad RX packet error	2:59:40 am Kernel panic not syncing:
4.37.31 pm Failed LCB components	8.52.37 pm Out of memory/Killed processes	3:00:00 am page_fault+0x1f/0x30
4.37.39 pm 2 nodes unavailable	8.55.13 pm Node unavailable	3:00:03 am Node unavailable
Failed within 12 min.	Failed within 11 min.	Failed within 16 min.

TBP Framework

The framework follows the standard division of steps for any machine learning model.

TBP Learning:

TBP uses TOT to learn the failure chains from the training data(Logs).

Node Failure Prediction:

TBP compares the incoming phrases with those in the failure chains. If chains with at least 50% similarity in log messages are formed, the corresponding node is likely to fail in the future.

The work flow

The main idea is that every phrase is assigned a topic.

We have finite number of topics for an integrated document.

During the training phase, TOT learns top N topics referring to phrases.

TBP forms sequences of phrases that correspond to failures in the past referring to the data. We use them to forecast future failures when those phrases reappear in the test data.

Topics Over Time

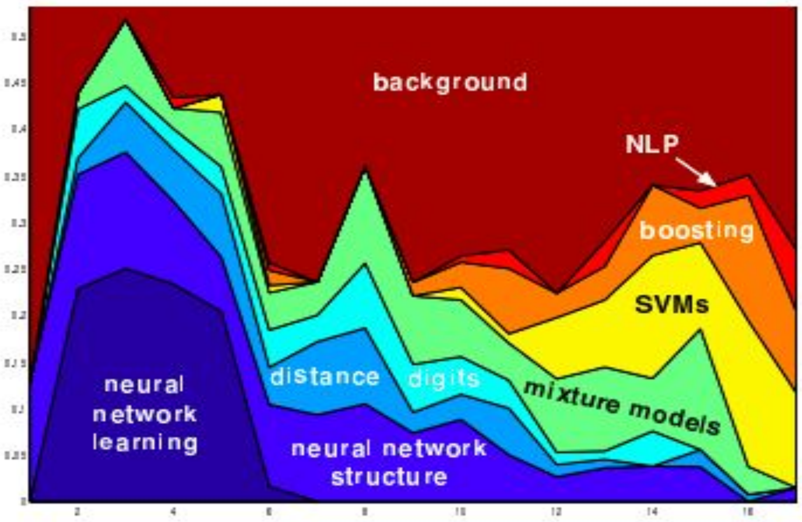
Topics over time captures the relationship between topic frequencies with respect to time.
It views time as a continuous entity and does not discretize time.

The intuition behind using TOT is that in a continuous and long running system like HPC systems, the topics evolve over time and reflect the state of the system at the current time period in consideration.

Capturing information from Logs

The requirement is to capture information in the form of correlations between highly probable topics at any given time.

Example:



Preprocessing Steps

Job Logs and Data Integration:
Logs corresponding to one event can show up across various places in the system. They are correlated using a timestamp difference of 15ms.

After successful correlation, a text document with timestamps, node ids and filtered log messages is formed.

Training Phase

Phrase Likelihood Estimation:

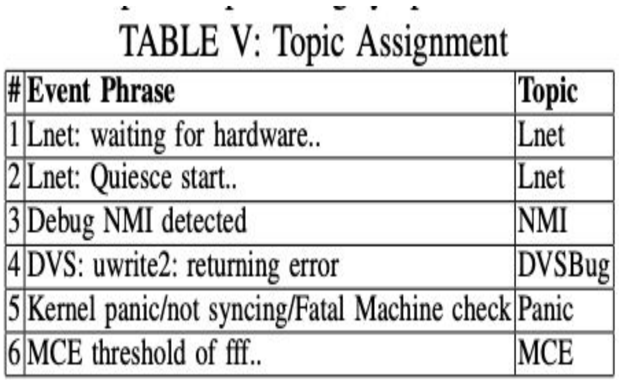
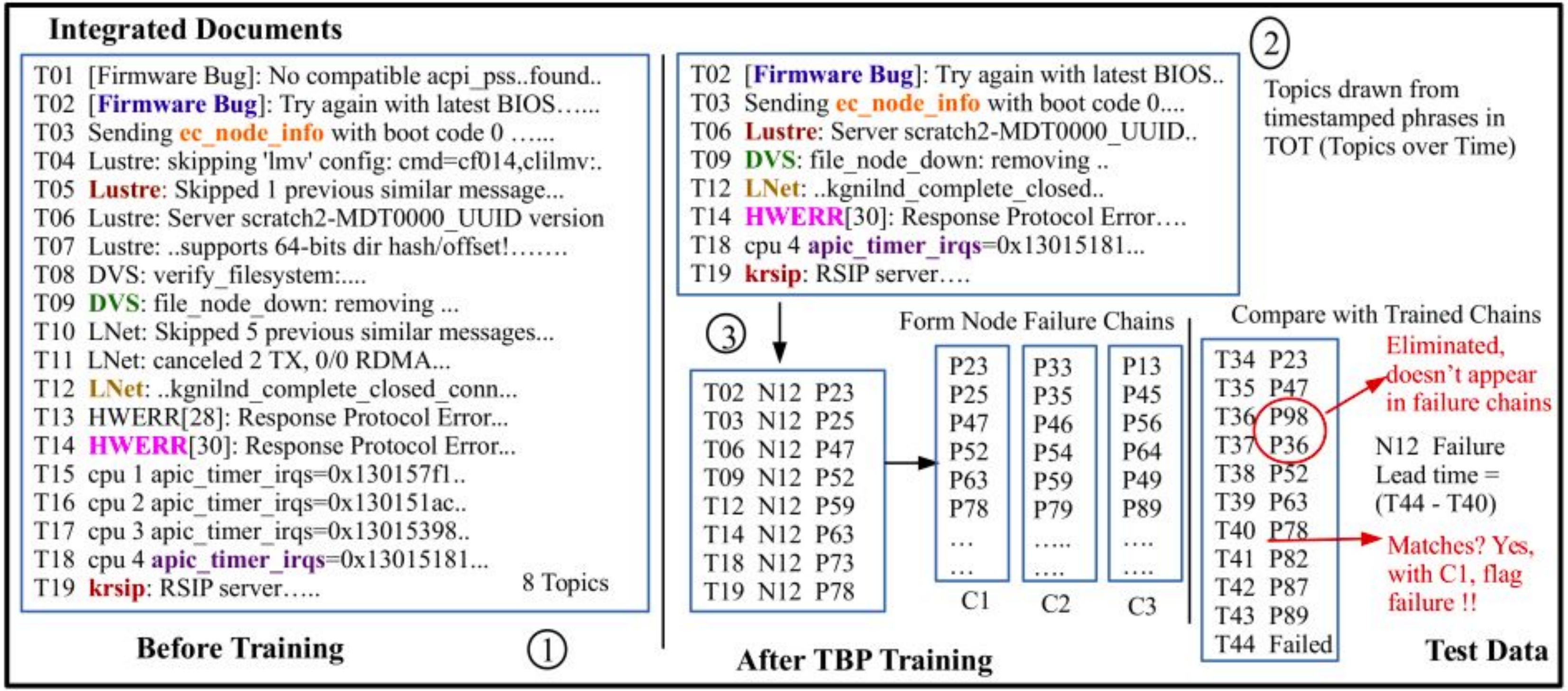


TABLE V: Topic Assignment

#	Event Phrase	Topic
1	Lnet: waiting for hardware..	Lnet
2	Lnet: Quiesce start..	Lnet
3	Debug NMI detected	NMI
4	DVS: uwrite2: returning error	DVSBug
5	Kernel panic/not syncing/Fatal Machine check	Panic
6	MCE threshold of fff..	MCE

The training phase includes topic assignment and identification of the top N topics over a period of time. This follows from a continuous time statistical technique called Topics over Time.



```
T01  [Firmware Bug]: No compatible acpi_pss..found..
T02  [Firmware Bug]: Try again with latest BIOS…...
T03  Sending ec_node_info with boot code 0 …...
T04  Lustre: skipping 'lmv' config: cmd=cf014,clilmv:.
T05  Lustre: Skipped 1 previous similar message...
T06  Lustre: Server scratch2-MDT0000_UUID version
T07  Lustre: ..supports 64-bits dir hash/offset!…….
T08  DVS: verify_filesystem:....
T09  DVS: file_node_down: removing ...
T10  LNet: Skipped 5 previous similar messages...
T11  LNet: canceled 2 TX, 0/0 RDMA...
T12  LNet: ..kgnilnd_complete_closed_conn...
T13  HWERR[28]: Response Protocol Error...
T14  HWERR[30]: Response Protocol Error...
T15  cpu 1 apic_timer_irqs=0x130157f1..
T16  cpu 2 apic_timer_irqs=0x130151ac..
T17  cpu 3 apic_timer_irqs=0x13015398..
T18  cpu 4 apic_timer_irqs=0x13015181...
T19  krsip: RSIP server…..
```

```
T02  [Firmware Bug]: Try again with latest BIOS..
T03  Sending ec_node_info with boot code 0....
T06  Lustre: Server scratch2-MDT0000_UUID..
T09  DVS: file_node_down: removing ..
T12  LNet: ..kgnilnd_complete_closed..
T14  HWERR[30]: Response Protocol Error….
T18  cpu 4 apic_timer_irqs=0x13015181...
T19  krsip: RSIP server….
```

②
Topics drawn from timestamped phrases in TOT (Topics over Time)
③

```
T02  N12  P23
T03  N12  P25
T06  N12  P47
T09  N12  P52
T12  N12  P59
T14  N12  P63
T18  N12  P73
T19  N12  P78
```

```
P23   P33   P13
P25   P35   P45
P47   P46   P56
P52   P54   P64
P63   P59   P49
P78   P79   P89
…     …..   ….
…     ….    ….
C1    C2    C3
```

```
T34  P23
T35  P47
T36  P98
T37  P36
T38  P52
T39  P63
T40  P78
T41  P82
T42  P87
T43  P89
T44  Failed
```

Eliminated, doesn't appear in failure chains
N12 Failure
Lead time =
(T44 - T40)
Matches? Yes, with C1, flag failure !!
Test Data

Fig. 5: TBP Prediction: Topic Modeling for Node Failure Prediction

Performance

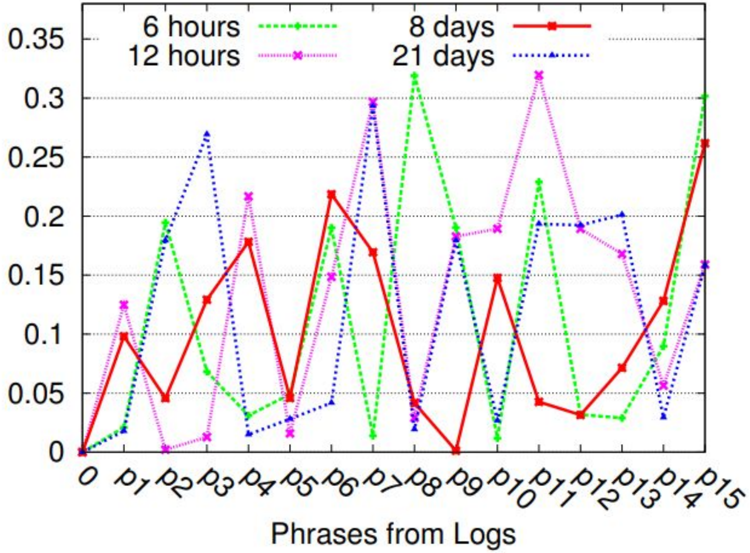
The data shows that node failures are actually somewhat rare, which calls into question the utility of TBP.

However, the number of compute node failures increases with service node failures; predicting service node failures will prevent cascading failures.

Also, rescheduling jobs after node failures is expensive; the job scheduler could avoid running long jobs on nodes with short term failure predictions.

Observation - Phrase distribution

There is significant phrase variation over a short time interval, which means that disparate, large events occur in the system with high frequency. As a result, discrete time models can’t be used here, because they cannot capture variation beyond their time granularity.



Prediction quality and lead time

In their experiments, TBP is trained on 4 weeks worth of logs and tested on a week's worth of data.

In this scenario, it predicts 86% of all node failures correctly. However, it needs to be retrained with 4 weeks worth of data every week to maintain its level of performance.

TBP offers at least a minute worth of lead time. This can be improved by pruning the failure event chains, at the expense of more false positives.

Thoughts

TBP does provide a novel method by taking into consideration the lead times, low level logs, continuous time environment.

The details about the application of TOT algorithm are not obvious.

Training phase requires manual intervention to establish correlation of logs.

Does this work for online learning?