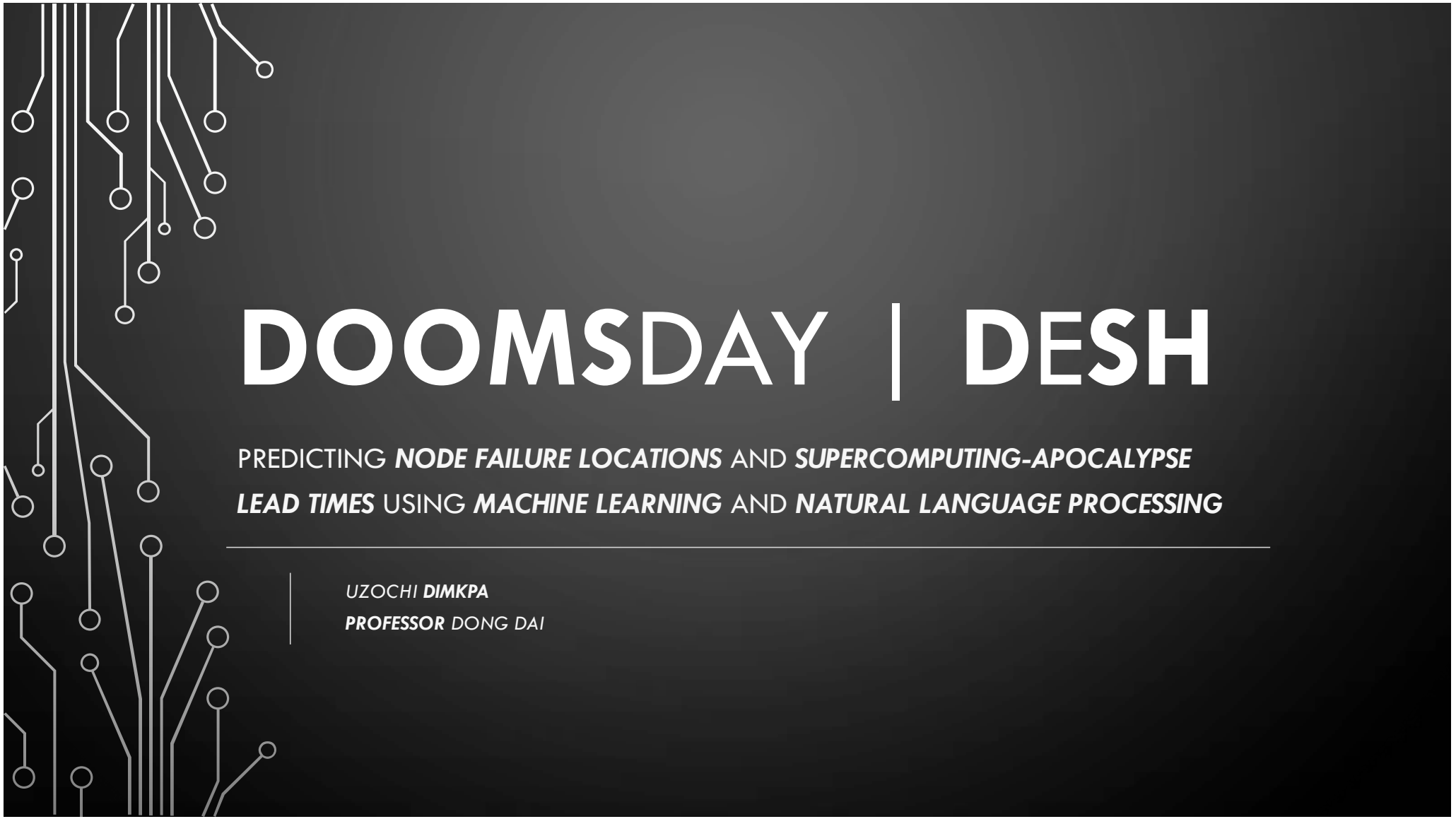


DOOMSDAY | DESH

PREDICTING ***NODE FAILURE LOCATIONS*** AND ***SUPERCOMPUTING-APOCALYPSE LEAD TIMES*** USING ***MACHINE LEARNING*** AND ***NATURAL LANGUAGE PROCESSING***

UZOCHI ***DIMKPA***

PROFESSOR *DONG DAI*

THE **PROBLEM** ITSELF

PREDICTING DOOMSDAY ON AN HPC SYSTEM

- **MTBF** is **decreasing** as **HPC** system **component count increases**

- **MTBF** = Mean time between failures; **doomsday** is happening more often as **computation speed** (throughput) **increases** and **latency decreases**
  - **Higher MTBF** = higher **fault rates** = more **wasted computation time** & **resources** = **more doomsday occurrences**
  - Something **must** be **done** to **help fix this**

PREDICTING DOOMSDAY ON AN HPC SYSTEM (CONT'D)

- **Doomsday** = Node failure(s)

- **Node failure** = abnormal node shutdowns caused by some system anomaly triggered by software *and*/or hardware
  - Some software issues cause hardware issues; vice versa

- **Anomalous node failure** = A node failing because of issue other than maintenance; normally, these are caused by hardware/software errors
  - Shutting down, heartbeat failure/unresponsive, etc.

PREDICTING DOOMSDAY ON AN HPC SYSTEM (CONT'D)

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.

Examples of log phrase and timestamp sequences leading to node failures

Source – **Doomsday**

PREDICTING DOOMSDAY ON AN HPC SYSTEM (CONT'D)

- **Prediction** is **done** using combination of **natural language processing** and **semantic analysis** on **timestamped failure chains** (**De**SH speaks about semantic analysis; **Dooms**day doesn't)
  - **Natural language processing** = Log data contains phrases; these phrases have various meanings in different compute state contexts
  - **Semantic analysis** = LSTM layers (consisting of RNNs) encode contextual (definitional) relationships between phrases; closer together in vector = higher correlation

- **Timestamped failure chain** = ordered combination of collated phrases (integrated document); **pulled** from **log data** (VERY IMPORTANT)

LOG DATA
&
WHY IT'S IMPORTANT

WHY LOG DATA?

- **Vast amount** of **information available**
- **All*** **information** is **relevant**
  - * All **selected** log information
  - * Physical environment info (**SEDC** logs) discarded

SEDC – System Environment Data Collection; temperature, voltage, etc.

WHY LOG DATA? (CONT'D)

TABLE II: Data Details

Source	Content
p0 directory	Internals of compute nodes
Boot Manager	Boot node messages
Log System	`rsyslog` messages
Power/State Logs	Component power and state information
Event Messages	Event router records
SMW Messages	System Management Workstation messages
HSN Stats	High Speed Network Interconnect logs
Job Logs	Batch job/application scheduler messages

Sources of most prevalent logs, and their content

Source – **Dooms**day

WHY LOG DATA? (CONT'D)

- Log **information/structure** is **vendor-specific**
  - **Cray** is **not** the **same** as **BlueGene/P**
    - **Cray** – Intel; **BlueGene/P** – IBM
- Log **sources vary**
  - Cray logs are pulled from **many different locations**; BlueGene/P logs sources are **dissimilar**

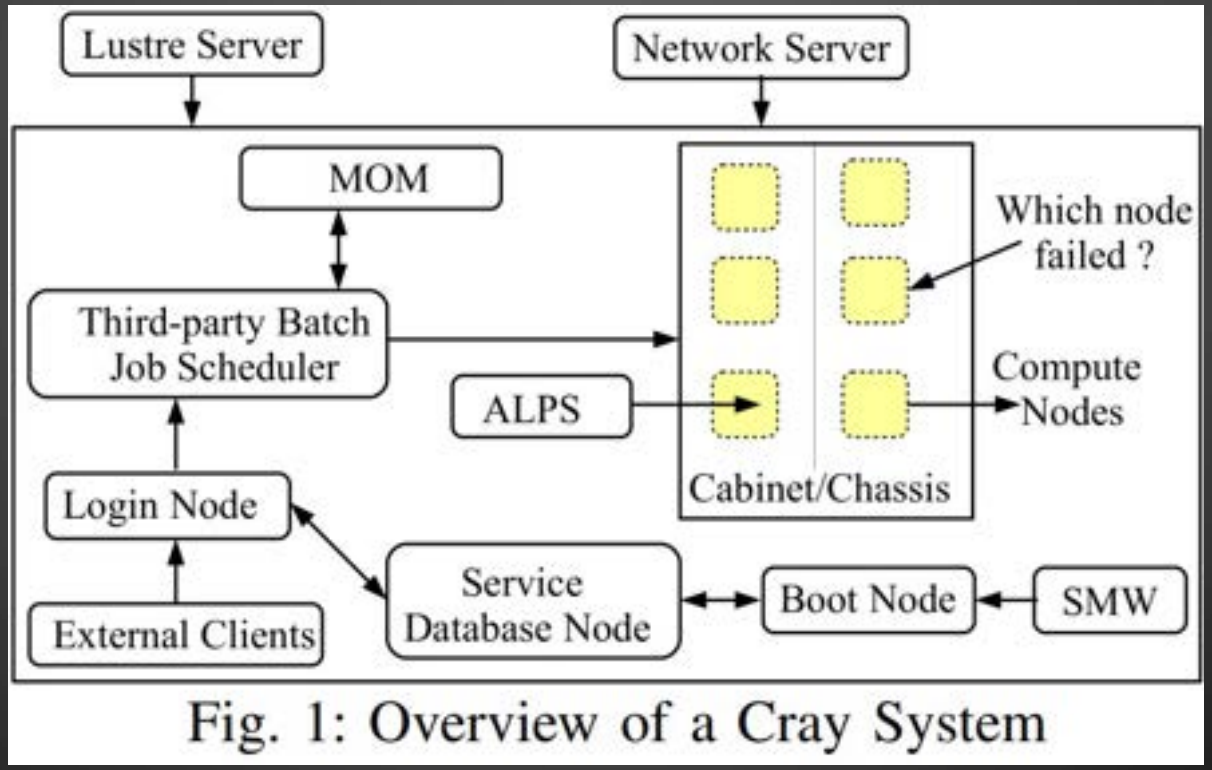


Fig. 1: Overview of a Cray System

High-level overview of Cray HPC system

Source – **Dooms**day

Lustre – Parallel Distributed File System
MOM – Machine Oriented Mini-server; handles job scheduling & execution on server
SMW – System Management Workstation; server administrator access location

```
[2008-07-08 03:10:01][c0-0c2s3n2]fsx-linux-aio[5706] general protection rip:421b60 rsp:7fffffffadc8 error:0
[2008-07-08 03:10:01][c0-0c2s3n2]fsx-linux-aio[5708]: segfault at fffffffffffffffb rip 0000000000421b60 rsp 00007fffffffae58 error 5
[2008-07-08 03:10:01][c0-0c2s3n2]fsx-linux-aio[5707]: segfault at fffffffffffffffb rip 0000000000421b60 rsp 00007fffffffae58 error 5
[2008-07-08 03:10:02][c0-0c0s7n3]fsx-linux-aio[7268] general protection rip:421b60 rsp:7fffffffadc8 error:0
[2008-07-08 03:11:13][c0-0c1s6n3]LustreError: 23331:0:(file.c:2415:ll_inode_revalidate_fini()) failure -2 inode 39221021
[2008-07-08 03:11:48][c3-0c2s0n0]Lustre: 13612:0:(llite_mmap.c:398:ll_nopage()) binary changed. inode 39246505
[2008-07-08 03:12:16][c0-0c1s4n2]Out of Memory: Kill process 2817 (memperf) score 699161 and children.
[2008-07-08 03:12:16][c0-0c1s4n2]Out of memory: Killed process 2817 (memperf).  apid: 307899
[2008-07-08 03:14:14][c0-0c2s3n0]LustreError: 5669:0:(file.c:2415:ll_inode_revalidate_fini()) failure -2 inode 39221286
```

Example Cray XT Console Log Data

```
080708 00:45:03  ############  ####  ############  ######  ######################  ######################  ####################
080708 00:45:03                Port  Remote        Remote  Recoverable             Send or Rcv
080708 00:45:03  Node ID       Num   Node ID       Port    Rcv Errs                Fatal Errs              Router Errs
080708 00:45:03  ############  ####  ############  ######  ######################  ######################  ####################
080708 00:45:03  c0-0c0s7s1    6     *             *                                                       Bad LUT Dir, Rsp Chan
080708 00:45:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
080708 00:53:03  c1-0c1s2s1    2     c1-0c1s3s1    3                                                       Deadlock Timeout, rsp1
080708 00:53:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
080708 00:53:03  c1-0c2s2s3    6     *             *                                                       Deadlock Timeout, Rsp Chan
080708 00:53:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
080708 00:53:03  c1-0c2s2s2    6     *             *                                                       Deadlock Timeout, Rsp Chan
080708 00:53:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
080708 00:53:03  c1-0c2s2s2    0     c1-0c2s2s3    5                                                       Deadlock Timeout, rsp1
080708 00:53:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
080708 00:53:03  c1-0c2s2s1    6     *             *                                                       Bad LUT Dir, Rsp Chan
080708 00:53:03  ------------  ----  ------------  ------  ----------------------  ----------------------  --------------------
```

Example Cray XT Netwatch Log Data

Netwatch – High-speed network traffic events

```
Tue Jul  8 04:01:22 2008 - rs_event_t at 0x805cd40
ev_id = 0x040040e5 (ec_heartbeat_stop)
ev_src = ::c0-0c2s6
ev_gen = ::c0-0c0s0n0
ev_flag = 0x00000002 ev_priority = 32 ev_len = 87 ev_seqnum = 0x00000000
ev_stp = 48732ce2.0002c0e2 [Tue Jul  8 04:01:22 2008]
svcid 0: ::c0-0c2s6n2 = svid_inst=0x0/svid_type=0x0/svid_node=c0-0c2s6n2[rsn_node=0x5a/rsn_type=0x0/rsn_state=0x7], err code 65740 - node heartbeat fault
ev_data...
00000000: 01 00 00 00 00 00 00 00 00 00 00 00 00 07 5a 00 *..............Z.*
00000010: 00 00 62 02 00 00 00 00 01 00 00 00 cc 00 01 00 *..b.............*
00000020: 6e 6f 64 65 20 63 30 2d 30 63 32 73 36 6e 32 20 *node c0-0c2s6n2 *
00000030: 6d 69 67 68 74 20 62 65 20 64 65 61 64 2c 20 6c *might be dead, l*
00000040: 61 73 74 20 68 65 61 72 74 62 65 61 74 20 30 30 *ast heartbeat 00*
00000050: 30 30 36 63 62 39 00                           *006cb9.........*
```

Example Cray XT Consumer Log Data

```
 26301146 KERN_0A26  KERNEL          _bgp_unit_dma       _bgp_err_dma_rec_fifo_not_avail         WARN     2009-01-24-18.51.52.192605 -                 0          - ANL-R07-M1-N00-256              R07-
M1-N04-J27                                              44V3575YL11K7262006 x'02402860101280A0912670D2CCAE' A DMA unit reception FIFO is full.  This is a recoverable error, but performance might be improve
d by increasing the FIFO size (via environment variable DCMF_RECFIFO=size-in-bytes) or be checking the DMA FIFOs more often.  Additional details:  1) torus location is (4,0,1)  2) Packet PID is 0.  3) rFIFO bit
 mask is 0b00000001.



 26301148 KERN_0803  KERNEL          _bgp_unit_ddr       _bgp_err_ddr_double_symbol_error        WARN     2009-01-24-18.55.09.389245 -                 0          - ANL-R43-M1-512                  R43-
M1-N00-J24                                              44V3575YL12M73389KK x'02401901000077 4C110A67C38CAD' ECC-correctable double symbol error: DDR Controller 1, failing SDRAM address 0x035340000, (1) BPC
 pin JB154, transfer 1, bit 129, BPC module pin O07, compute trace MEMORY1DATA128, DRAM chip U31, DRAM pin C8.(2) BPC pin HZ150, transfer 1, bit 132, BPC module pin N04, compute trace MEMORY0DATA129, DRAM chip
U26, DRAM pin C2.



 26301149 KERN_0809  KERNEL          _bgp_unit_ddr       _bgp_err_ddr_rbs_activated              WARN     2009-01-24-18.55.09.597319 -                 0          - ANL-R43-M1-512                  R43-
M1-N00-J24                                              44V3575YL12M73389KK x'02401901000077 4C110A67C38CAD' DDR Redundant Bit Steering activated on bit 0x00000081, chipselect 0x00000000, controller 0x00000
001
```

Example BlueGene/P RAS Log Data

Consumer – Log of all events on Cray Event Router

RAS – Reliability, Accessibility, Serviceability; contains information about system and OS environment

DOOMSDAY V.S. **DESH**

- From here, their log-data manipulation and integrated-document collation differ:
  - **Dooms**day –
    - Backwards pruning
    - Timestamps are correlated during all of training
    - Unknown phrases are virtually ignored
  - **DeSH** –
    - Phrase severity labeling
    - Phrase vector data splitting (static/dynamic)
    - Unknown phrases are lightly examined

DOOMSDAY V.S. **DESH** (CONT'D)

- **Unknown phrases**
  - Difficult to handle; their **impact** on **node failures** are **unknown**
  - e.g. **correctable MCE** – occurrence is so uncommon, # of topics chosen should be ≥ 150 in order to prevent TBP from discarding it as non-salient
  - **DeSH** does much more

TABLE XII: Difficult Correlation Extraction

#	Error	**Description**
1	Console	interrupt took X ns
2	Console	DVS: lnet_mapuvm: page count mismatch
3	Job	Node id has a different configuration
4	Console	logged... correctable MCEs

Examples of difficult phrases to correlate with node failures

Source – **Dooms**day


DOOMSDAY
(THE FIRST PAPER)

DOOMSDAY'S SOLUTION: **TBP**

- **TBP** – Time-Based Phrases
- A **phrase extraction scheme;** phrase **likelihood estimation** pulled from **continuous time-series data** to find **useful log phrases**

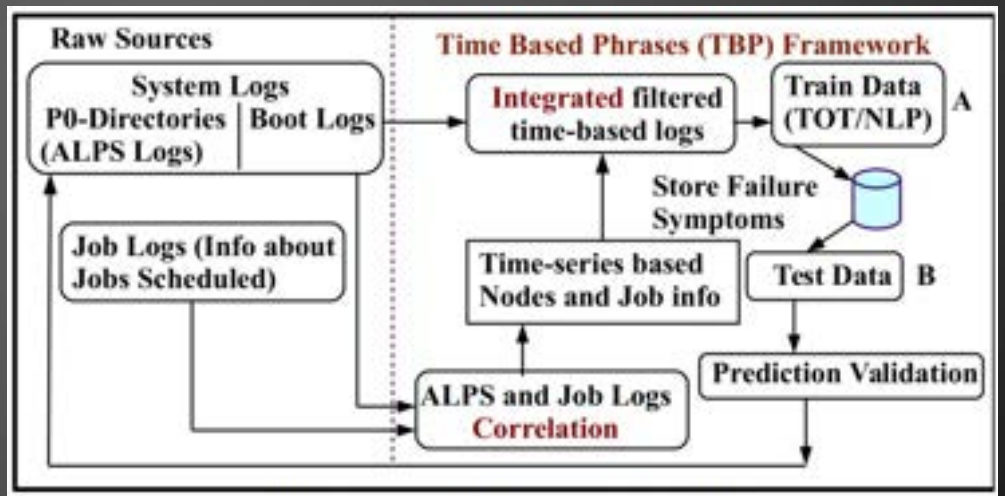


Fig. 3: TBP Framework

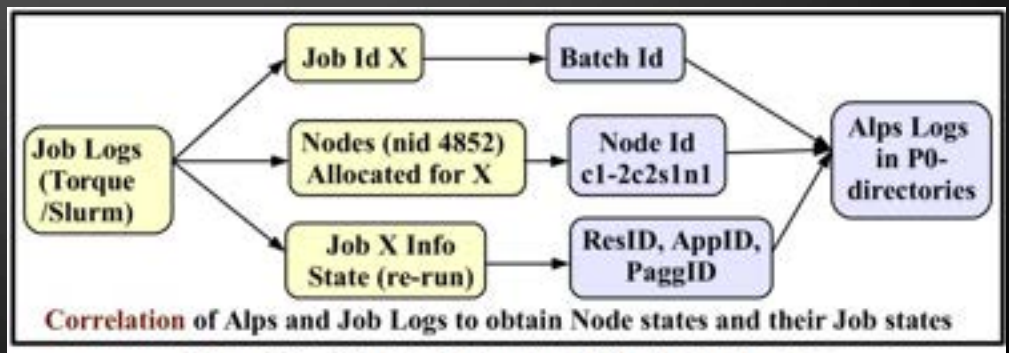


Fig. 2: Correlation with Job Logs

DOOMSDAY'S SOLUTION: **TBP** (CONT'D)

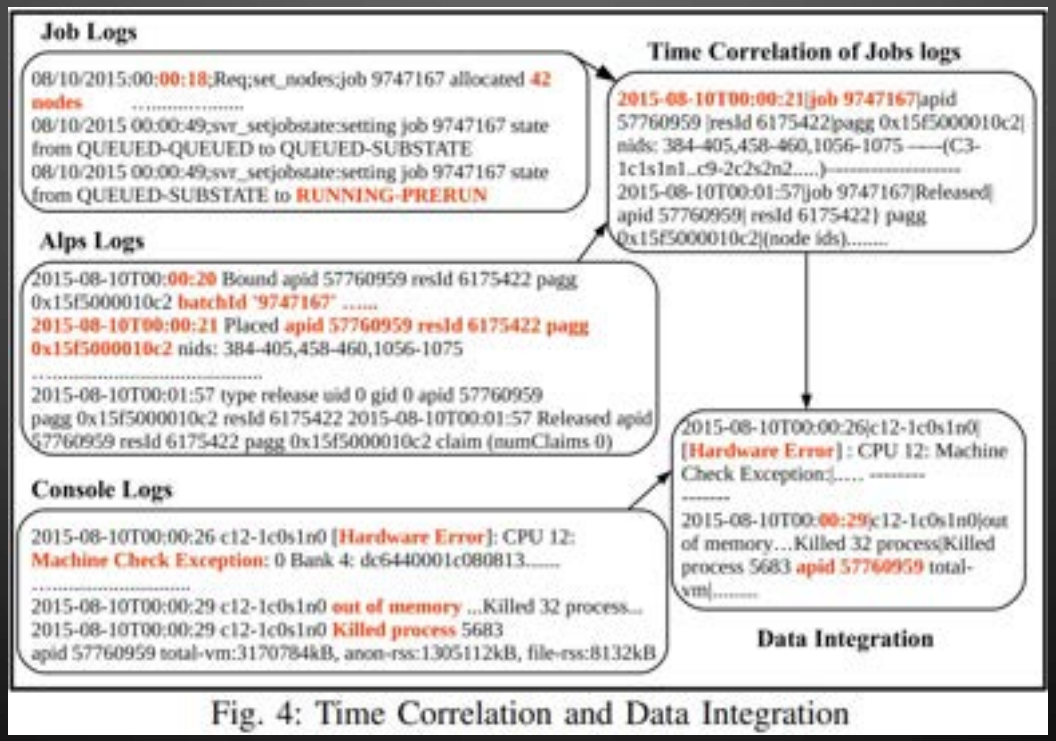


Fig. 4: Time Correlation and Data Integration

- **Data** & **timestamp correlation** process

ALPS – Application-Level Placement Scheduler; resource management software designed to work across multiple nodes running independent OS instances

DOOMSDAY'S SOLUTION: **TBP** (CONT'D)

- Employs **TOT** – Topics over Time
  - A **NLP technique;** identifies top N topics appearing in logs, then tracks how they change over time
  - Uses **Gibbs Sampling** – MCMC (Monte-Carlo Markov-Chain) algorithm
    - Obtains a **sequence of observations** from **multivariate data** – in this case**, integrated log data**
    - Used when **direct sampling** is **difficult**

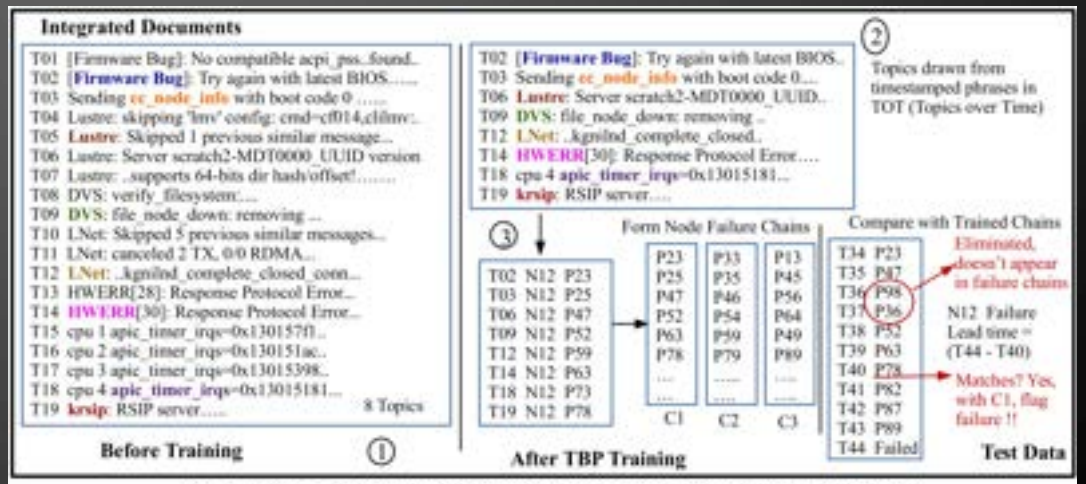
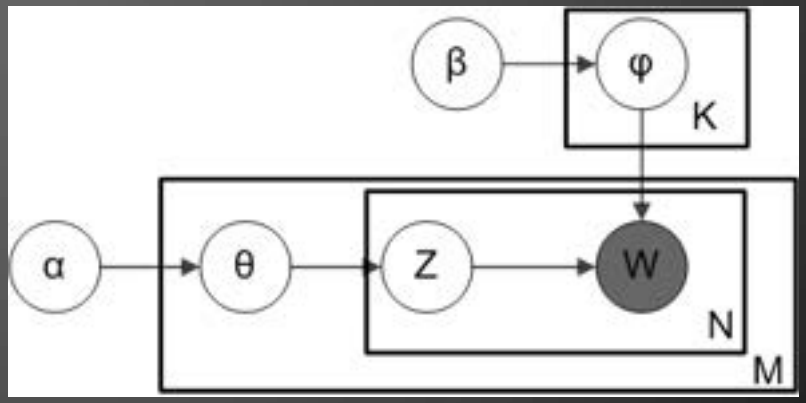


Fig. 5: TBP Prediction: Topic Modeling for Node Failure Prediction

DOOMSDAY'S SOLUTION: **TBP** (CONT'D)

- **TOT** uses **LDA**
  - **LDA –** Latent Dirichlet Allocation; **unsupervised learning**
  - Three-level hierarchical **Bayesian model**
  - **K** = # of topics
- Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words
  - An algorithm for grouping words under topics and topics under documents [3, 4]



M denotes the number of documents

N is number of words in a given document (document *i* has N_i words)

α is the parameter of the Dirichlet prior on the per-document topic distributions

β is the parameter of the Dirichlet prior on the per-topic word distribution

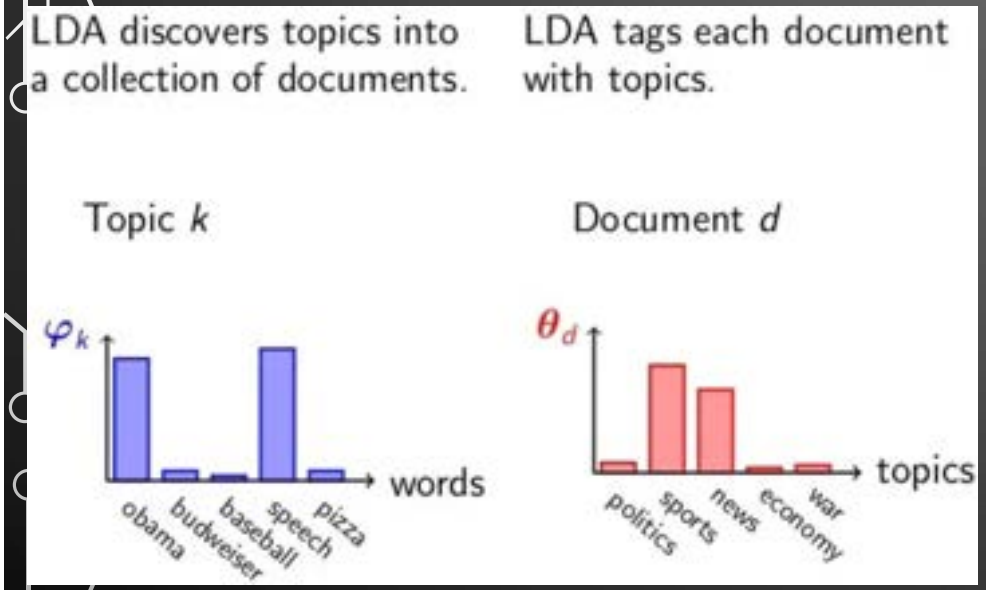
θ_i is the topic distribution for document *i*

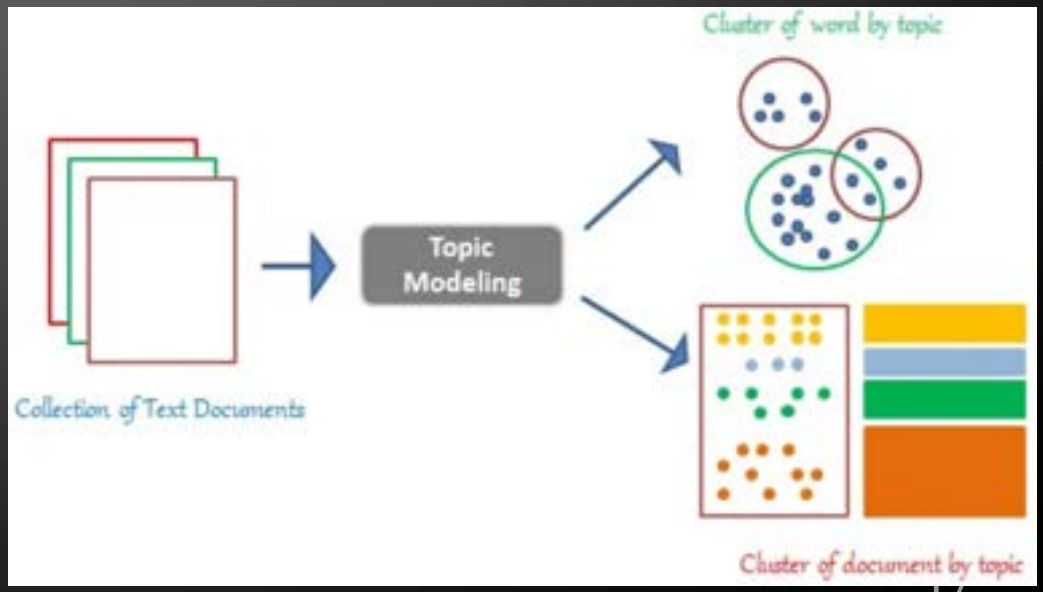
φ_k is the word distribution for topic *k*

z_{ij} is the topic for the *j*-th word in document *i*

w_{ij} is the specific word.

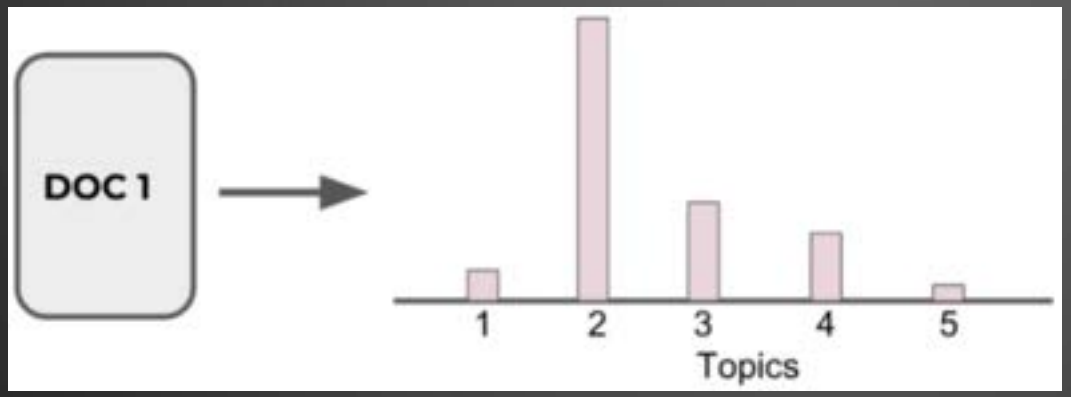
LATENT DIRICHLET **ALLOCATION**

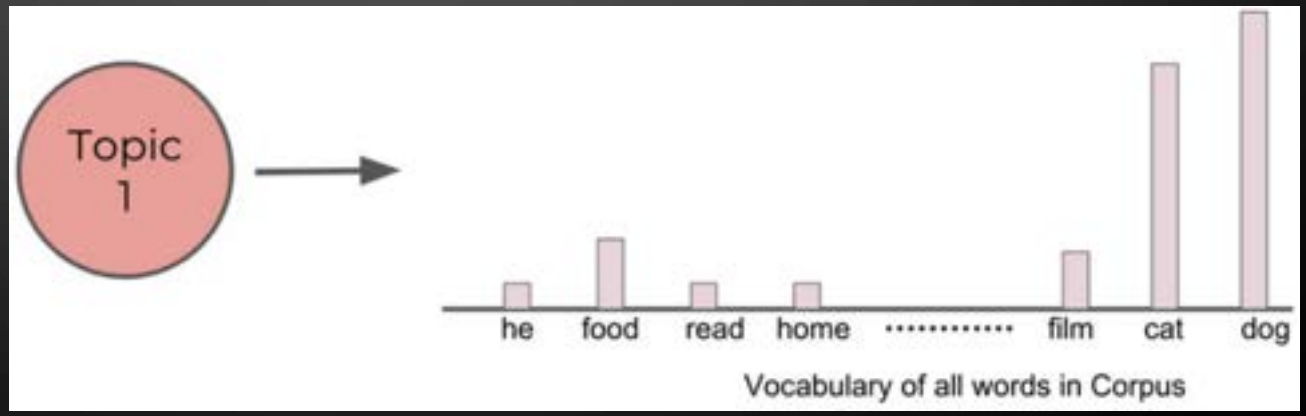




Source – **Medium**

LATENT DIRICHLET **ALLOCATION** (CONT'D)





Source – **Medium**

DOOMSDAY'S PHRASE **REDUCTION: BACK**-PRUNING

- Increases **prediction lead times** and **performance**
- Enacted when **enough** of the **failure chain** has been **seen** in the **testing data** (≥ 50%)
  - **20% pruning** – ignore last 20% of phrases; failure flagged
  - **30% pruning** – ignore last 30% of phrases; failure flagged

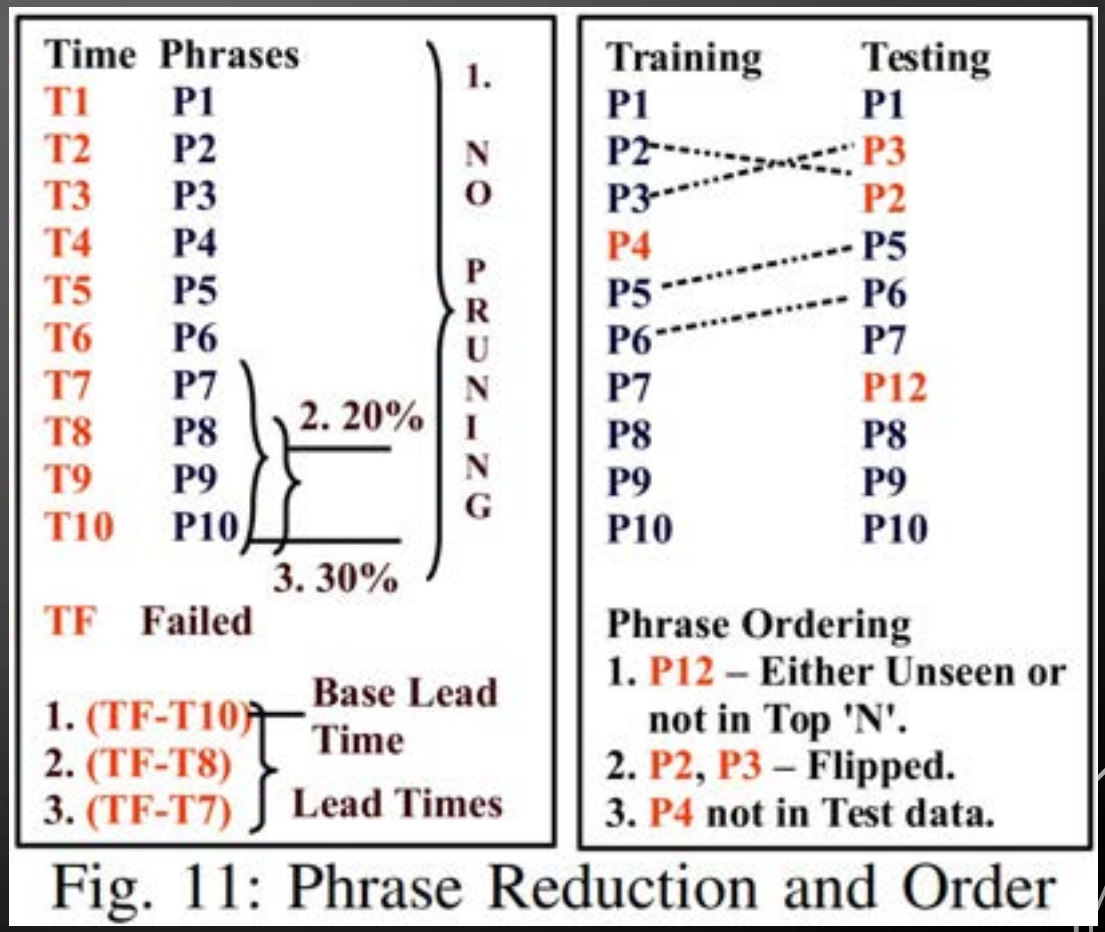


Fig. 11: Phrase Reduction and Order

DOOMSDAY'S **TBP** RESULTS

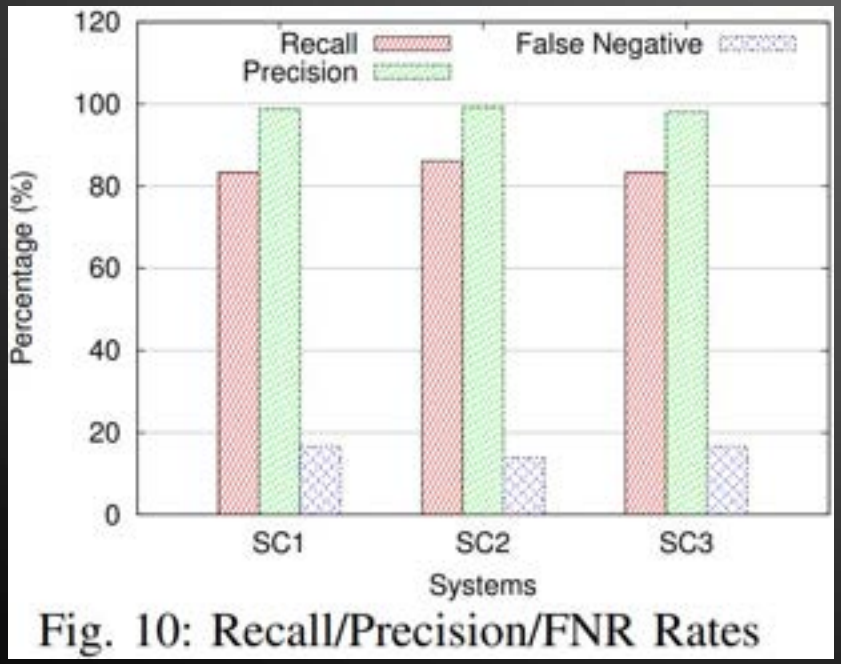


Fig. 10: Recall/Precision/FNR Rates

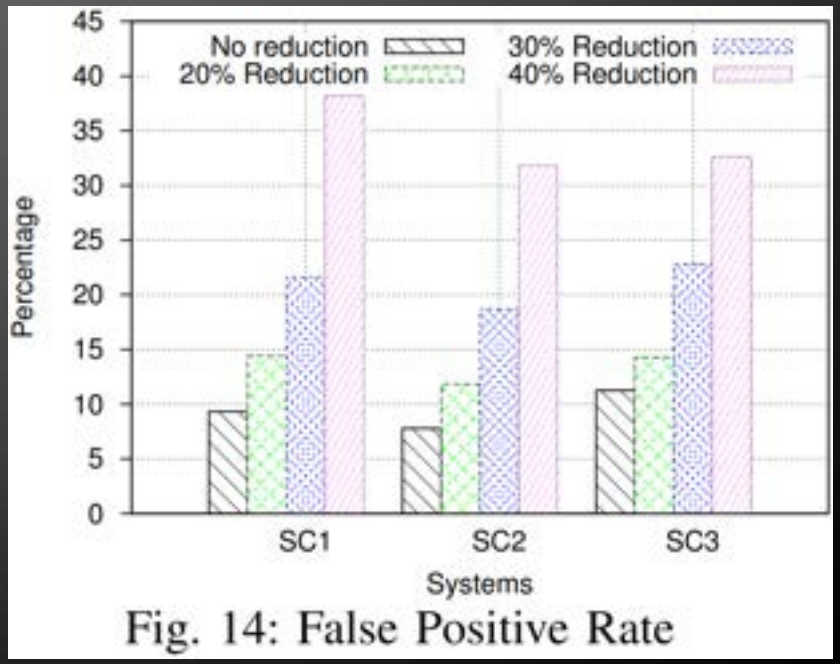


Fig. 14: False Positive Rate

DOOMSDAY'S **TBP** RESULTS (CONT'D)

Prediction **lead time increases** as **back-pruning** threshold **increases**

- 0.5 min – 0% back-pruning
- 1.1 min – 20% ``
- 1.6 min – 30% ``
- 4.2 min – 40% ``

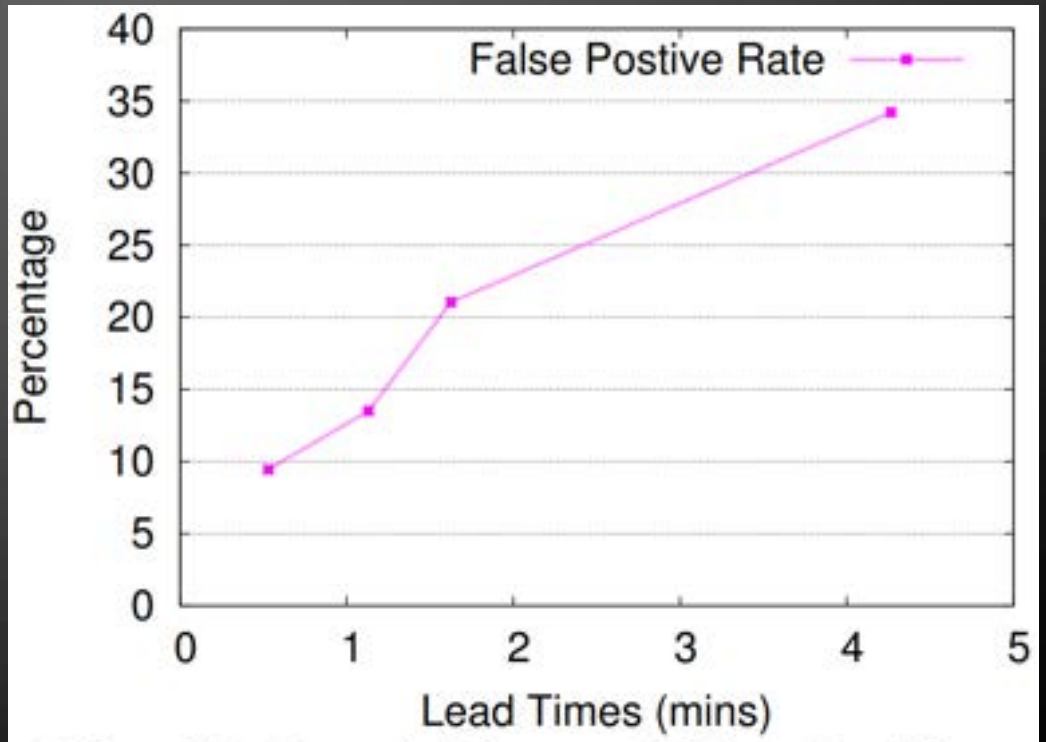


Fig. 13: Lead Times+False Positives

DOOMSDAY'S **TBP** VARIOUS METRICS

Fig. 12: Failure Categories

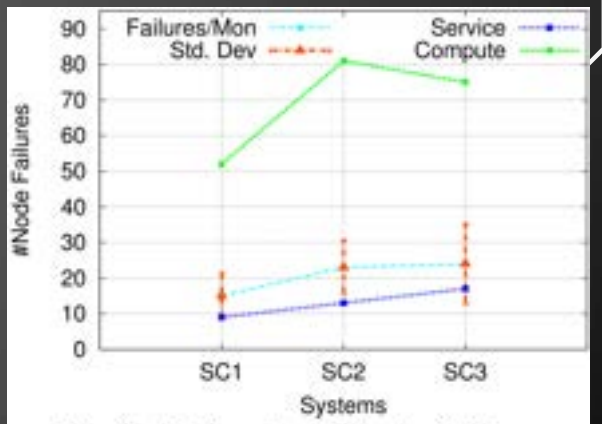


Fig. 6: Estimate of Node Failures

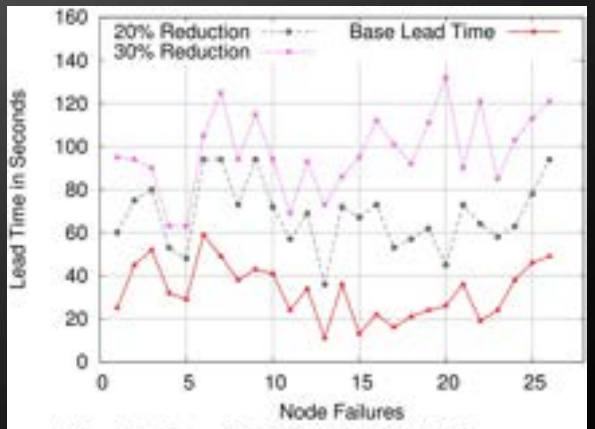


Fig. 9: Sensitivity of Lead Times

DOOMSDAY'S **TBP** VARIOUS METRICS (CONT'D)

TABLE VII: Recurring Phrases

No.	Phrases
P1	crms_wait_for_linux_boot: nodelist: *
P2	Lnet: Quiesce start: hardware quiesce
P3	Wait4Boot: JUMP:KernelStart *
P4	krsip:RSIP server * not responding
P5	startproc: nss_ldap: failed to bind
P6	checking on pid *
P7	LustreError: *:*:.....can't find the device name
P8	GNII_SMSG_SEND + *
P9	No .....bios_settings file found
P10	Lnet: Added LNI *
P11	DVS: file_node_down: removing *
P12	Lustre: skipped * previous similar messages
P13	Lnet: skipped
P14	<node_health:*> RESID* xtnhc FAILURES
P15	Bad RX packet error

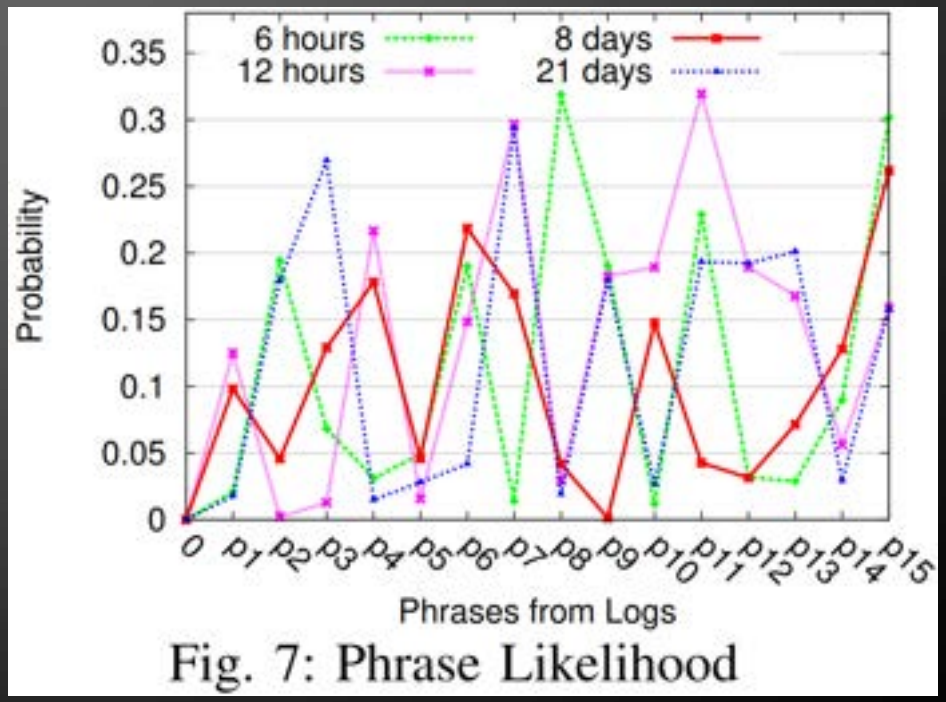


Fig. 7: Phrase Likelihood


DESH
(THE SECOND PAPER)

DESH – THE SOLUTION

- A stacked LSTM made up of multiple RNN layers
- Three phase training/testing scheme
  - 1) Trains failure chains per node; concatenates all failure chains
  - 2) trains delta-time/timestamps with failure chains per node; concatenates all failure chains
  - 3) Tests entire LSTM model on each node

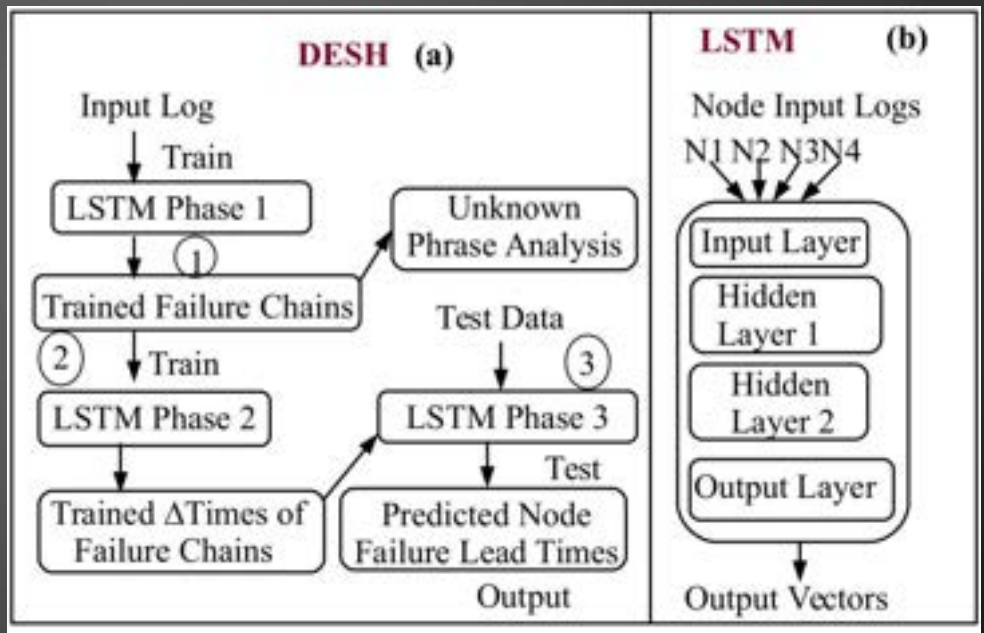


Figure 1: Desh with LSTM

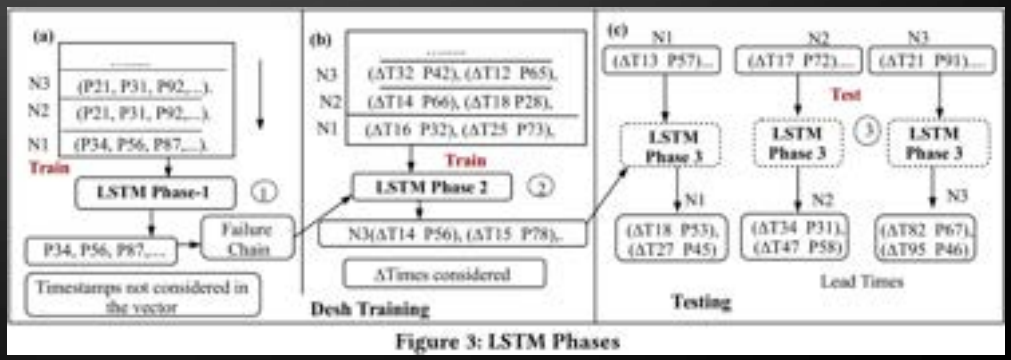


Figure 3: LSTM Phases

DESH – THE SOLUTION (CONT'D)

- The **same LSTM model** is updated each time
- After training, model-encoded information includes
  - **Phrase inter-relational semantics** – observes **phrase distributions across vectors** as **distance** from one another (within vector)
  - General proportions of **safe/unknown/error** phrase messages to expect in node failures
  - **Pertinent phrase information** contributing to node failures

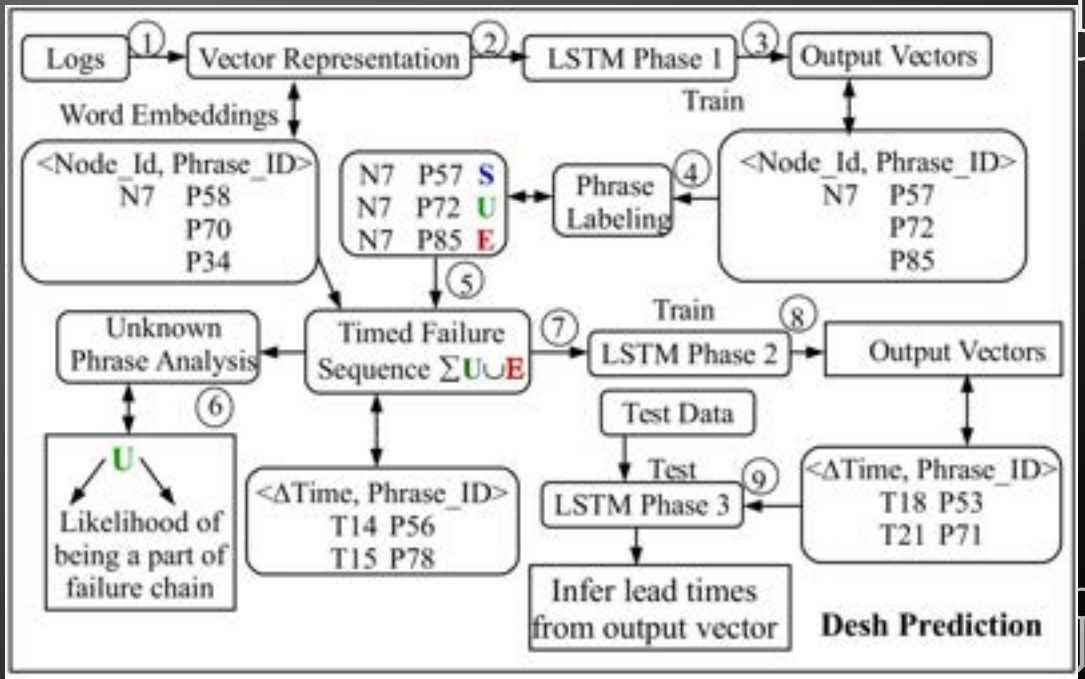


Figure 2: Desh Overview

DESH – THE SOLUTION (CONT'D)

- LSTM **structure** & **hyperparameters**
  - **HL** – hidden layers
  - **Steps** - # of upcoming phrases to predict
  - **HS** – history size; # of phrases to consider in testing data output before making prediction
- Each layer is an **RNN** – encodes **short-term variations** effectively

Table 5: LSTM Parameter Specifications

#	Input Vector	Output Vector	#HL	Steps	#HS	Loss Function, Optimizer
Phase-1	(P1, P2..PN)	(P11, P15..PN)	2	3	8	SGD, categorical crossentropy
Phase-2	(Δ T1, P1), (Δ T2, P2,..)	(Δ T11, P11), (Δ T22, P22,..)	2	1	5	MSE, Rmsprop
Phase-3	(Δ T4, P4), (Δ T5, P5,..)	(Δ T15, P15), (Δ T16, P16,..)	2	1	5	MSE, Rmsprop

DESH – THE SOLUTION (CONT'D)

- Static & dynamic phrase info
  - **Dynamic** discarded; **time-dependent**
  - **Static** kept; **time-independent**
- Three phase training/testing scheme

Table 2: Phrase Vectors

#	Timestamp (T)	Node Id (N)	Phrase (P)	
			Static	Dynamic
1	16:25:48.301744	c1-0c1s1n0	kernel * LNet: hardware quiesce *	p0-20141216t162520, All threads awake
2	16:39:59.507009	c4-0c0s0n2	Running * using values from *	sysctl, /etc/sysctl.conf
3	00:01:16.704832	c2-0c0s15n2	hwerr * Correctable aer replay timer timeout error*	[28451]:0x6624, Info1=0x500: Info2=0x18:
4	10:47:39.417963 (T1)	c0-0c0s0n2 (N1)	hwerr *:ssid rsp a status msg protocol err error* (P1)	:Info1=0x4c00054064: Info2=0x0: Info3=0x2

Table 4: Example Failure Chain

#	Timestamp	Phrase	Label	Phrase Vector
P1	03:59:58.466 (T1)	CPU *: Machine Check Exception:	U	ΔT1=07.822, P1
P2	03:59:59.543 (T2)	[Hardware Error]: Run the above through 'mcelog –ascii	U	ΔT2=06.745, P2
P3	04:00:00.477 (T3)	[Hardware Error]: RIP !INEXACT! 10:	U	ΔT3=05.811, P3
P4	04:00:01.706 (T4)	Kernel panic - not syncing: Fatal Machine check	E	ΔT4=04.582, P4
P5	04:00:01.731 (T5)	Call Trace:	E	ΔT5=04.557, P5
P6	04:00:06.288 (T6)	cb_node_unavailable	E	ΔT6=00:000, P6

DESH'S RESULTS

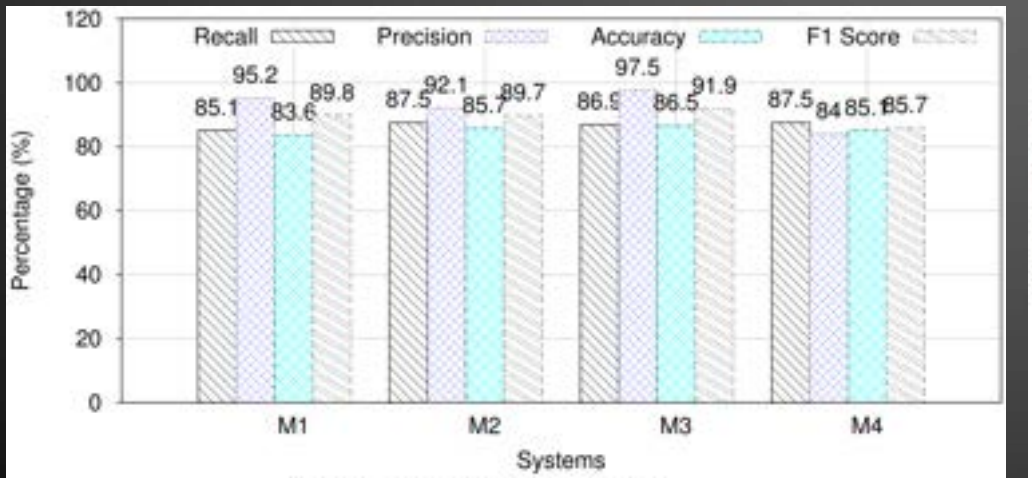


Figure 4: Prediction Rates

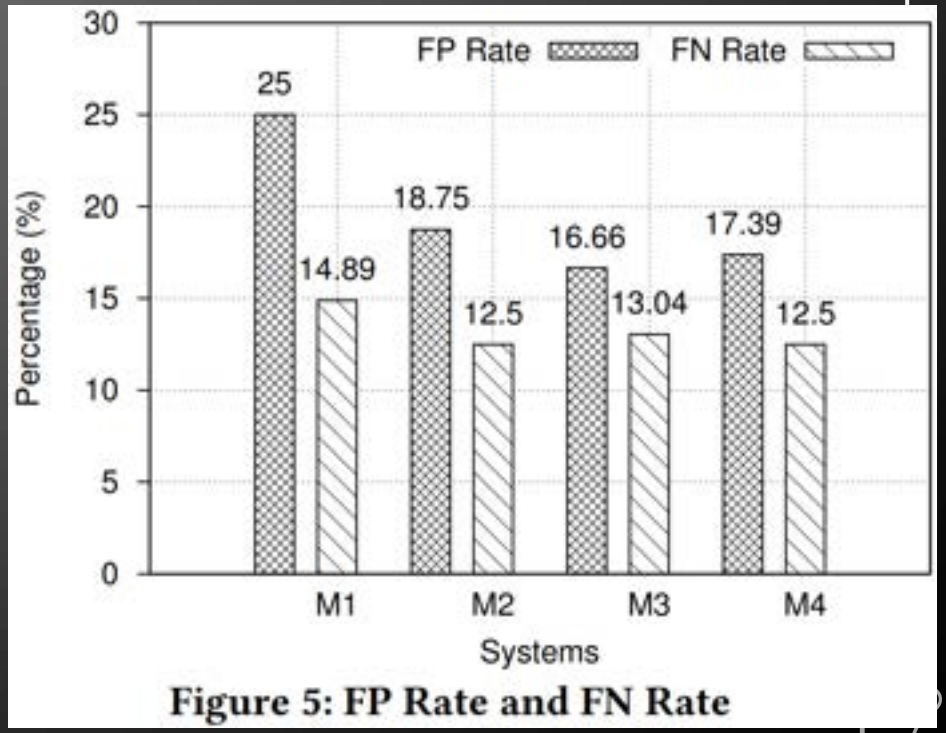


Figure 5: FP Rate and FN Rate

DESH'S RESULTS (CONT'D)

Major point of **trade-off:**

False alarm rate

v.s.

Avg. lead time

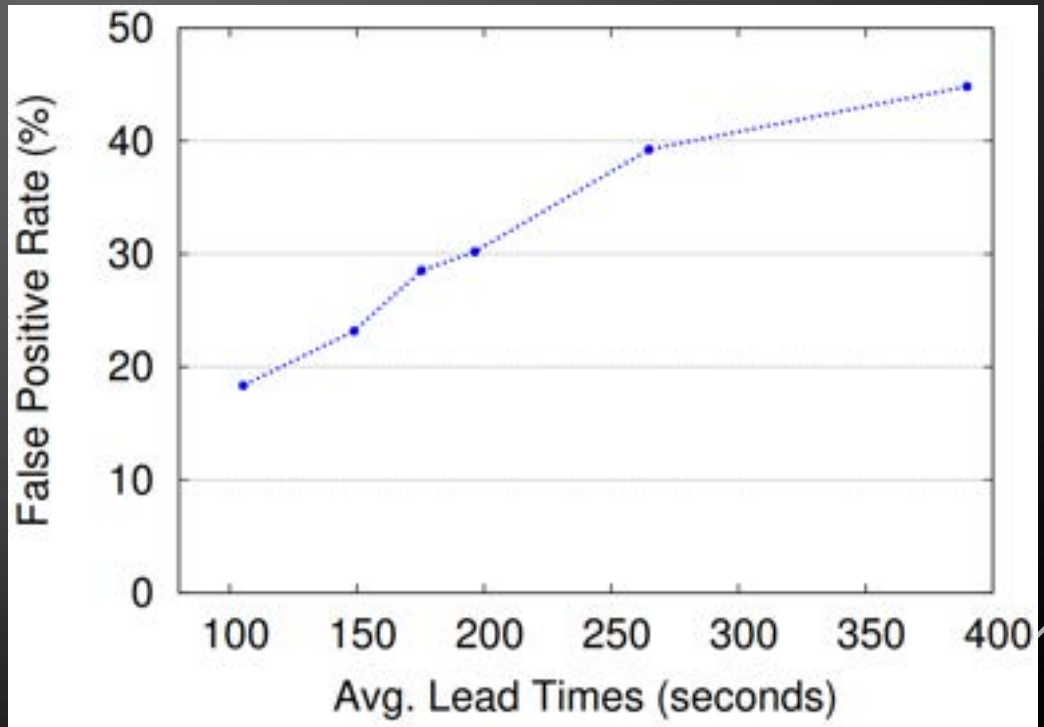


Figure 8: Lead Times and FP Rate

DESH'S VARIOUS METRICS

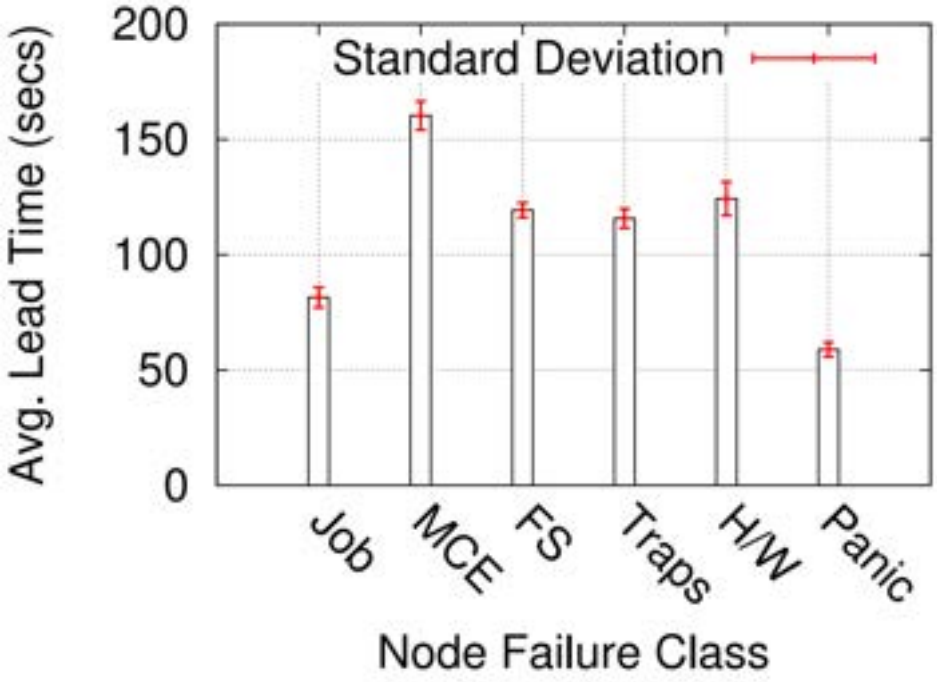


Figure 6: Lead Times+Failure Classes

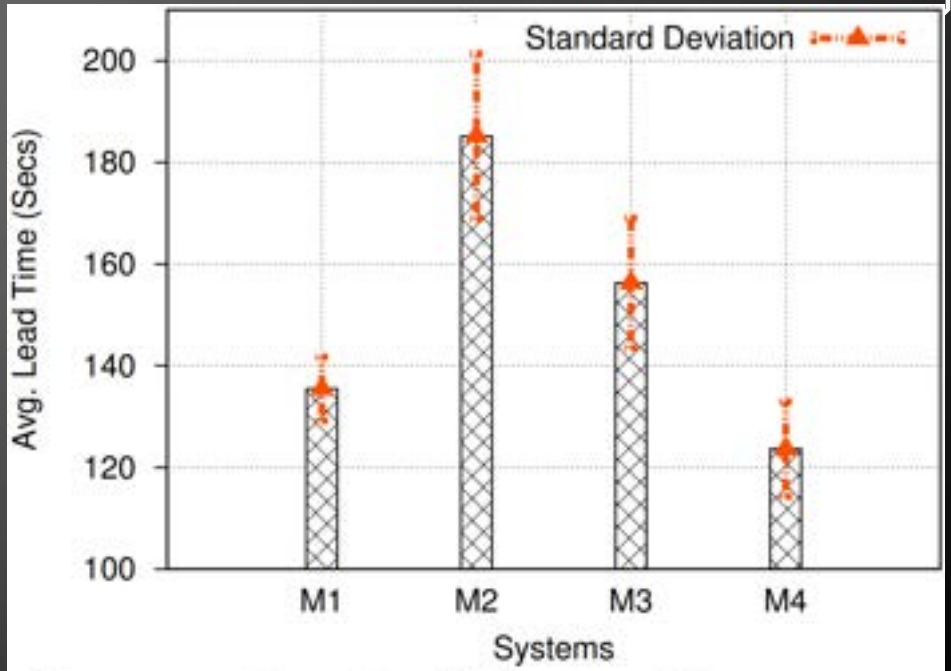


Figure 7: Avg. Lead Times of Systems

DESH'S VARIOUS METRICS (CONT'D)

Major point of **trade-off:**

Computation time

v.s.

History Size &

# of steps

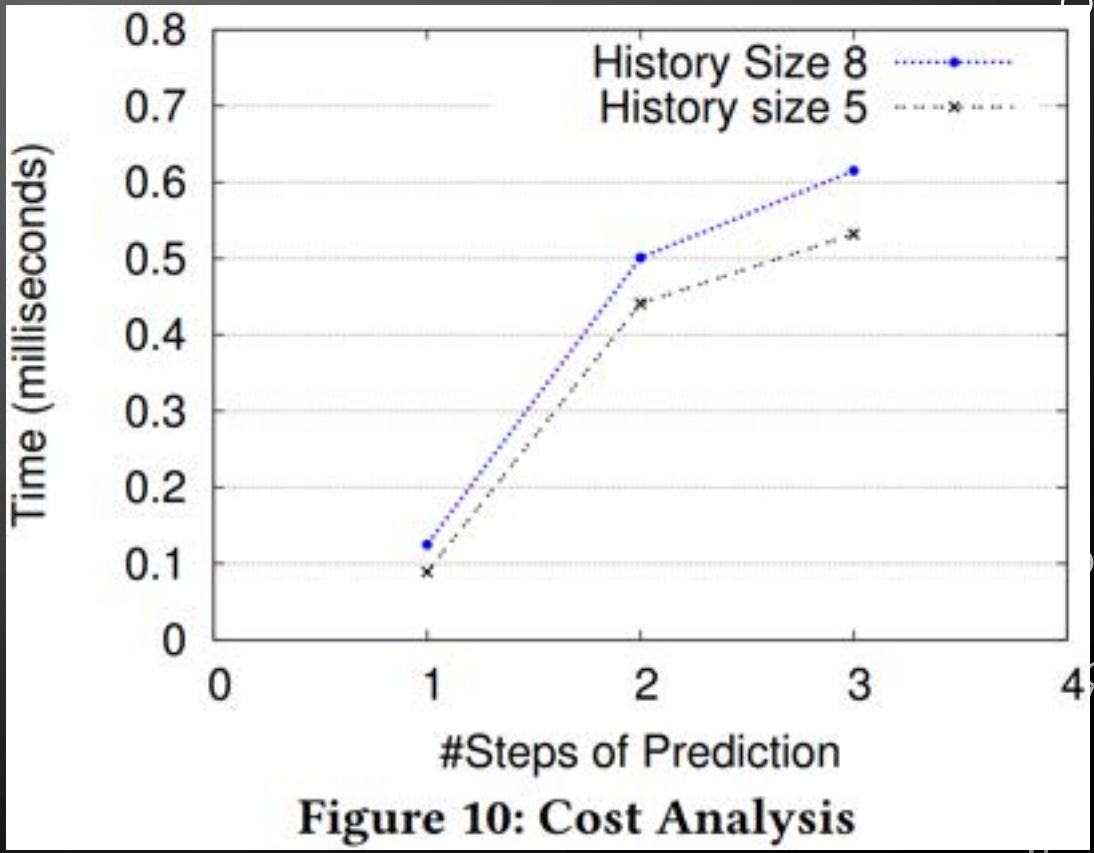


Figure 10: Cost Analysis

DESH – HANDLING **UNKNOWN** PHRASES

Table 8: Unknown Tagged Phrases

#	Phrase	(%)
P1	LustreError *	56
P2	Out of Memory/Killed Process	15
P3	Lnet: Critical H/W error	36
P4	Slurm load partitions error: Unable to contact slurm controller	42
P5	hwerr[*]: Correctable AER_BAD_TLP Error *	12
P6	Sent shutdown to llmrd at process *	17
P7	AER: Multiple corrected error recvd *	21
P8	Trap invalid code * Error *	8
P9	modprobe: Fatal: Module * not found *	27
P10	<node_health> * Warning: program * returned with exit code *	29
P11	DVS: Verify Filesystem	60
P12	BUG: unable to handle kernel NULL pointer dereference	25

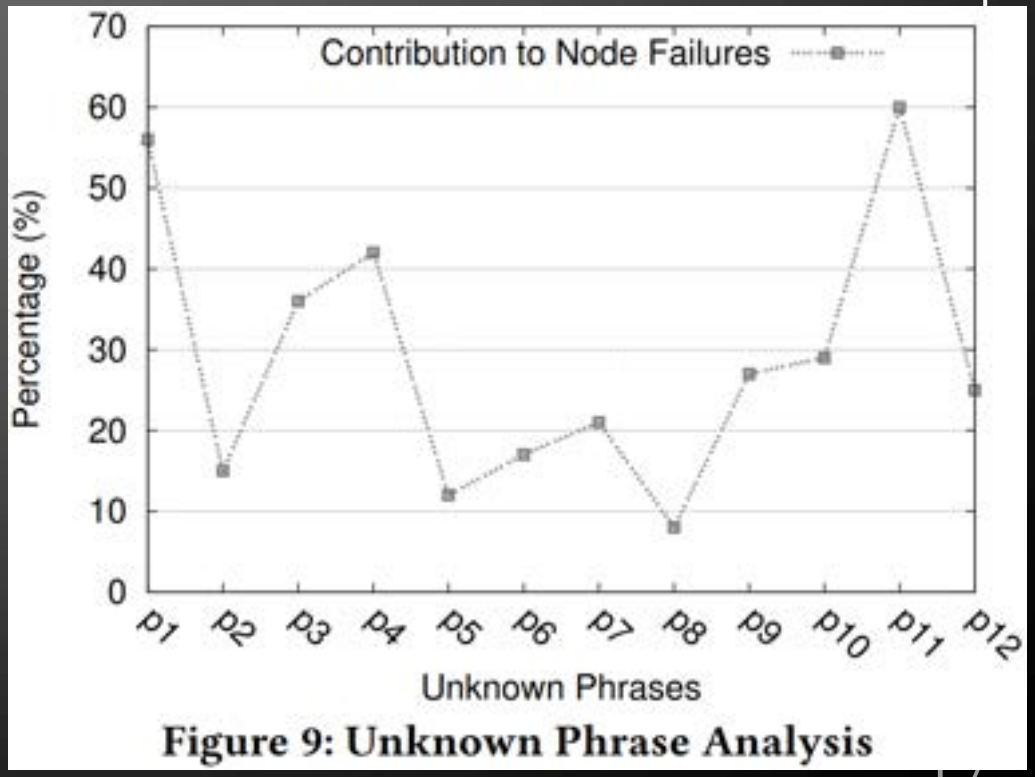


Figure 9: Unknown Phrase Analysis

DESH – HANDLING **UNKNOWN** PHRASES (CONT'D)

Table 9: Unknown Phrases with and without Node Failures

#	Failure 1	Failure 2	Not Failure 1	Not Failure 2
1	H/W Error: MCE Logged	LustreError *	nscd: nss_ldap reconnected	LustreError: Skipped
2	Corrected Memory Errors on Page *	DVS: Verify Filesystem: *	<node_health> program * returned with exit code *	nscd: nss_ldap reconnected
3	<node_health> program * returned with exit code *	DVS: * no servers functioning properly	Trap Invalid Code	Hw Error: MCE Logged
4	mce_notify_irq: *	Startproc: nss_ldap: failed..	Killed process *	Corrected DIMM Memory Errors
5	Lnet: critical hardware error: *	Stop NMI Detected	Out of memory *	MCE_notify_IRQ
6	[Gsockets] debug [0]: critical h/w error	Slurm load partitions error: Unable to contact slurm controller	Lustre: * binary skipped *	Lnet: H/W Quiesce
7	Stop NMI Detected	Slurmd Stopped	hwerr[*]: RSP A_status_msg_protocol_error*	Corrected Memory Errors on Page *
8	<node_health> warning: * node is down	System: halted	<node_health> * failures: The following tests * failed	Lustre: * connected to *

- Example log phrase sequences; not failure chains
- **Unknown phrases** occur in both **benign** and **failure** chains
  - Unknown phrases by themselves DO NOT contribute to node failures; **unknown phrases** in **context** to other phrases DO **contribute to node failures**

TRIGGERING PROACTIVE/**PREEMPTIVE MEASURES**

PROACTIVE **MEASURES**

Depending on error:
Can be triggered
between **20 seconds and 2 minutes** in **advance**

- They include the following:
  - Efficient (lazy) checkpointing
    - There are studies being conducted to **optimize checkpointing timing**
  - Node quarantining & process/job migration
    - Can **avoid 5 – 9%** of node failures if nodes are quarantined
    - Migration takes between **0.29 – 24.4 seconds** on average

PROACTIVE **MEASURES** (CONT'D)

- They include the following (Cont'd):
  - Root cause diagnosis (NOTE: **extremely complex**)
  - Node cloning
    - **Saves** on **redundant executions**
  - Preemptive node examination
    - Lowers MTTI

The key:

lower false alarm rate

MTTI – Mean Time to Interrupt; how long on average it takes an application to finish executing or be interrupted

RELATED **WORK**

At time of Doomsday paper, this work was **mostly the first of its kind;** most (if not all) other papers addressed the same issues with **variations in solution approaches**

TABLE XIII: TBP Comparison

Solutions	Method	Lead Time	Recall	System	Location
Hora [34]	Bayesian Networks	10 mins	18%	Dist. RSS Feed Reader	Component specific
Zheng+ [21]	Genetic Algorithms	10 mins	60%	Blue Gene/P	Rack-level
Li+ [7]	SVM, KMeans	10 mins	N/A	Blue Gene/P, Glory	Component with sensor
hPrefects [9]	Stochastic Model, Clustering	N/A	N/A	256 node HPC cluster	H/W, S/W components
[20]	Decision Tree (DT)	N/A	80%	HPC (LANL)	Node-level
[17]	Neural-gas [36]	N/A	N/A	Blue Gene	Node-level
[35]	SVM/DT/MLP etc.	17, 22 mins	91.34%	HPC cluster	Node-level
TBP	Topic Modeling	2 mins	86%	Cray	Node-level

Neural Gas – an unsupervised neural network topology learning algorithm that finds general, multilabel data classifications from feature vectors; a generalization of k-means; can classify feature vectors within multiple labels

Source – **Dooms**day

Table 10: Desh Comparison

Solutions	Method	Lead Time	Recall	Precision	Anomaly Injection	System	Location
Hora [38]	Bayesian Networks	10 mins	83.3%	41.9%	✓	Dist. RSS Feed Reader	component specific
Gainaru et al. [21]	Signal Analysis	N/A	60%	85%	×	Blue Waters	N/A
Islam et al. [29]	Deep Learning	N/A	85%	89%	×	Google Cluster	Job-level
UBL [14]	Self-Organizing Map (SOM)	50 secs	N/A	N/A	✓	RUBiS, Hadoop, System S	N/A
CloudSeer [45]	Automatons, FSMs	N/A	90%	83.08%	✓	OpenStack	N/A
Desh	Deep Learning	3 mins	86%	92.2%	×	Cray	node-level

Source – **DeSH**

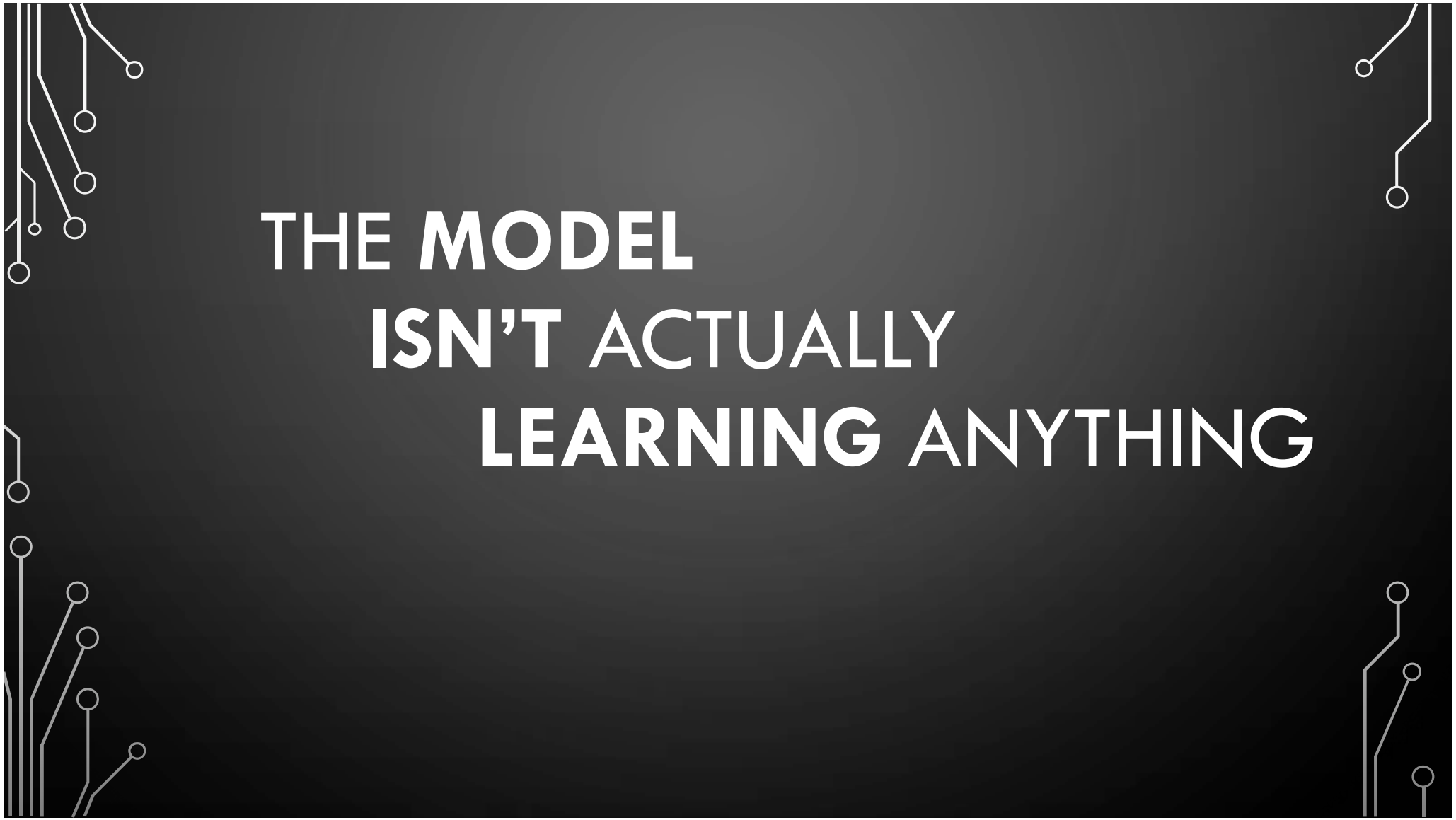
Both use **stacked LSTM**

Other solutions label & **augment logs** from source; **DeSH** does not

Table 11: Desh vs. DeepLog

#	Features	Desh	DLog
1	No Source-Code	✓	✓
2	Lead Time	✓	×
3	Component location	✓	×
4	Sequence-level Anomaly	✓	×
5	Injected Failures	×	✓
6	Node Failures	✓	×
7	Cloud+HPC	×	✓
8	False Positive Rate	✓	×

Source – **DeSH**


THE **MODEL**
ISN'T ACTUALLY
LEARNING ANYTHING

WHY IS NO TRUE LEARNING HAPPENING?

- Has **no idea** what any of the **log phrases** actually **mean**
  - The log phrases **actually do mean something;** that's **not** what the model is **learning**

- Only **'memorizes'** (learns) what **log event(s)** to **expect** next from a given node

WHY IS NO TRUE LEARNING HAPPENING? (CONT'D)

- **Unable** to **diagnose** actual **node issue**
- Can **only flag** a **failure** before it occurs and then **trigger** counteractive/preventative **measures**
- "system **anomaly** #/type ≥ threshold = impending **node failure**"
  - This is **all** that the model **really knows**

SOURCES

1. A. Das, F. Mueller, P. Hargrove, E. Roman and S. Baden, "Doomsday: Predicting Which Node Will Fail When on Supercomputers," SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, Dallas, TX, USA, 2018, pp. 108-121, doi: 10.1109/SC.2018.00012.
2. Das, Anwesha & Mueller, Frank & Siegel, Charles & Vishnu, Abhinav. (2018). Desh: Deep Learning for System Health Prediction of Lead Times to Failure in HPC. 40-51. 10.1145/3208040.3208051.
3. Blei, David M., et al. "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, JMLR, Jan. 2003, https://jmlr.org/papers/volume3/blei03a/blei03a.pdf.
4. https://medium.com/analytics-vidhya/latent-dirichelt-allocation-1ec8729589d4


THANK
YOU!


QUESTIONS?