Failure data-driven selective node-level duplication to improve MTTF in High Performance Computing Systems

Nithin Nakka, Alok Choudhary
EECS Department
Northwestern University
Chicago, IL. USA

Motivation

- MTTF or MTBF is a commonly used metric for system reliability
  - Length of time for which the system can be expected to work without any failures
- Used to determine checkpointing interval or mission time
- Understanding failure behavior can greatly enhance fault tolerance and reliability techniques
  - Runtime fault-tolerance and reliability techniques determine the MTTF of the system

Understanding System Failure behavior

- Failure and repair logs are valuable source of field failure information
- The granularity of the logs determines the extent to which they can aid reliable design
  - System-level logs can help improve system level techniques
  - Node-level techniques can be used to enhance node-level techniques

A key observation

- "All nodes are not equal"
  - By functionality or By failure behavior
- Node level reliability information
  - Can guide node level techniques such as duplication towards most critical portions of the system
    - Decreases system setup, maintenance and performance costs
  - Can help in customizing the fault tolerance techniques to the reliability requirements of the application
    - Used to customize Node-level duplication to meet MTTF requirements of an application

Contributions of the work

1. Analysis of node-level failures and their correlation with network topology and physical locations of the nodes in the system
2. Data-driven estimation of the coverage provided for selective duplication of nodes in the system
3. A methodology for selecting a small subset of the nodes that need to be duplicated to achieve the MTTF requirements of the application

Systems under study

- 22 supercomputing systems from Los Alamos National Laboratory under study
- Failure and usage information collected for over 9 years
- Systems have been categorized into 8 types (***A*** through ***H***) depending on nature of individual nodes – processor, memory etc.
- Systems have *smp*, *cluster* or *numa* architectures

Data trace format

- The failure trace is created by system administrators who detected and repaired failures
- Some important fields in the failure log are:
  - *System:* the number of the system being traced
  - *nodenumz:* The number of the node in the system
  - *Problem started*: Date and time at which problem was detected
  - *Problem fixed*: Data and time at which problem was fixed and system restored to original state
  - *Root cause*: Hardware, Software, Facilities, Network, Human error, or Unknown.

Our approach

- Failure data specific to a single node in a system is collected
- Records are ordered in increasing order of time of occurrence – "*Prob started*" field
- TTF for a fault $i = TTF_i = (prob\ started)_i - (prob\ started)_{i-1}$
- $$MTTF = \frac{\sum_{i}^{n} TTF_i}{n} = \frac{(prob\ started)_n - (system\ install\ date)}{n}$$
- TTR for a fault $i = TTR_i = (prob\ fixed)_i - (prob\ started)_i$
- $$MTTR = \frac{\sum_{i}^{n} TTR_i}{n}$$

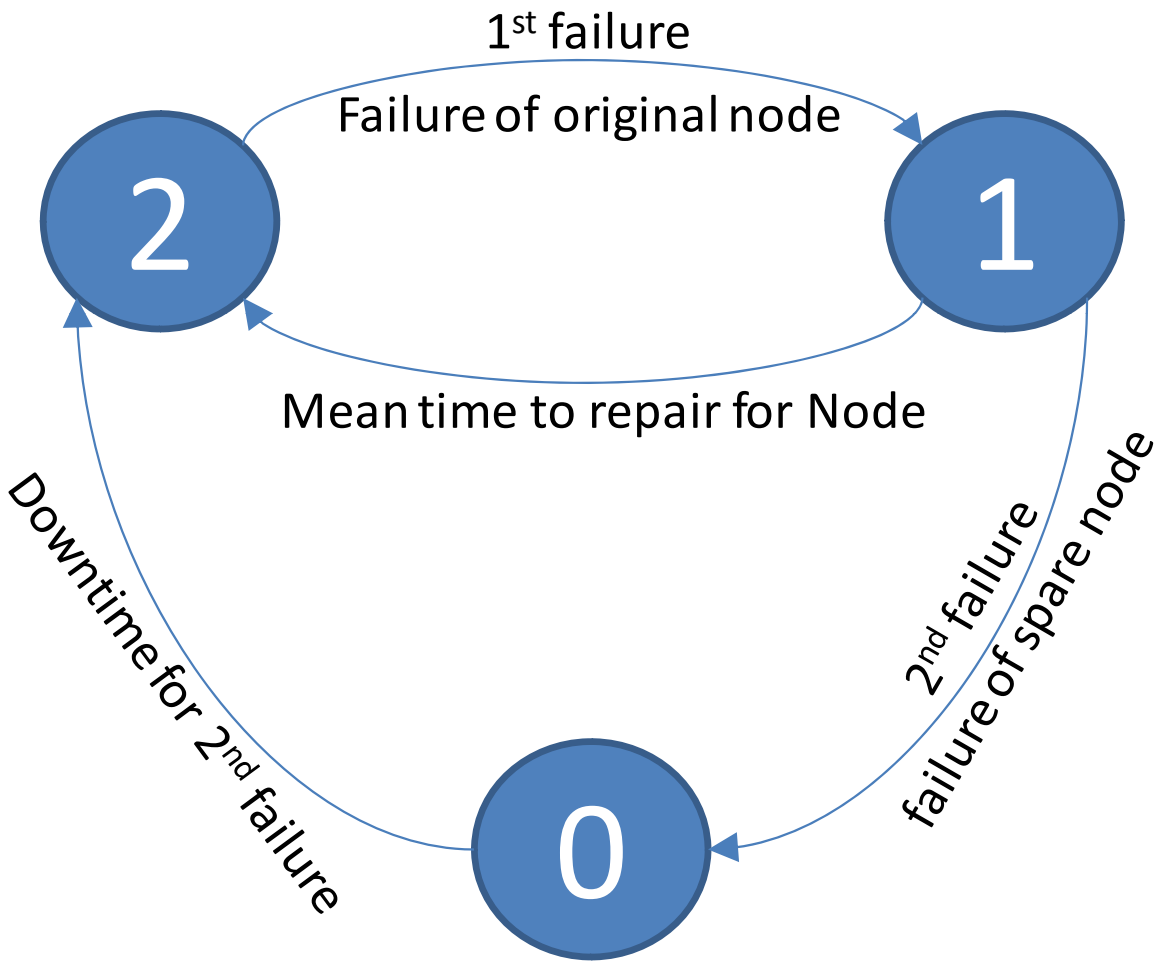
Node-level duplication

- A node is augmented with a spare node
  - Node failure is detected and computation is instantaneously transferred to spare node
  - Original node is repaired and is ready for normal operation in time
  - System is restored to original state and duplication can tolerate this failure
- If a second failure occurs before the original node is repaired duplication cannot tolerate this failure

$$(prob\,started)_i + (MTTR) > (prob\,started)_{i+1}$$

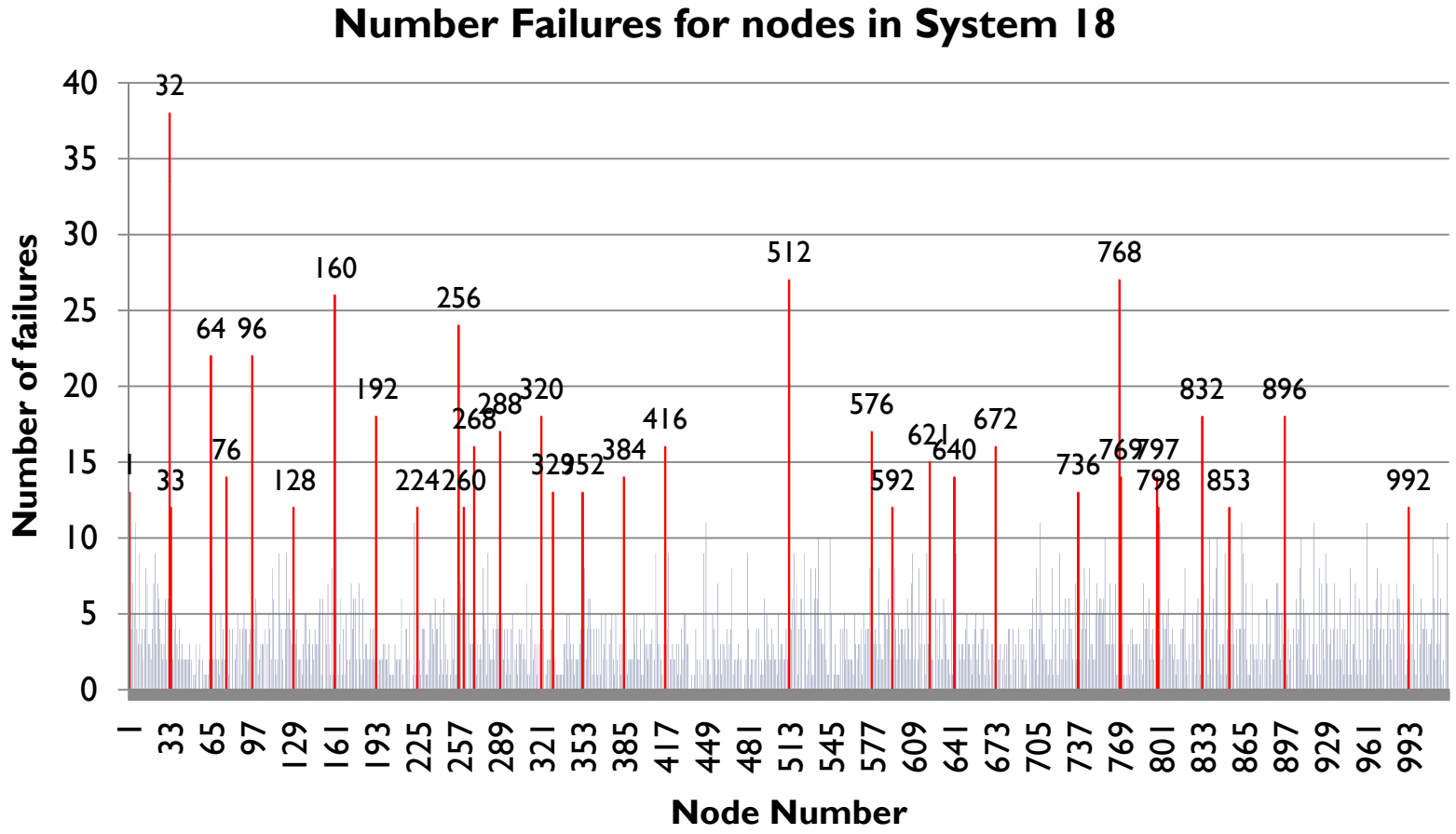
- MTTF for the system is recalculated with the number of faults remaining after duplication (n)

Duplication State Diagram



Failure rate vs. network configuration

- Nodes within a system show variability in failure rate due to network configuration



System network configuration

- Plot for number of failures shows peaks at nodes numbered as a multiple of 32 (with a few exceptions)
- Discussion with sys admin revealed that
  - Systems are configured into groups of 32 nodes with the 32nd node serving as an NFS root for the other 31 nodes in the group
  - Additional root type services make workload considerably higher than the rest hence failing more often

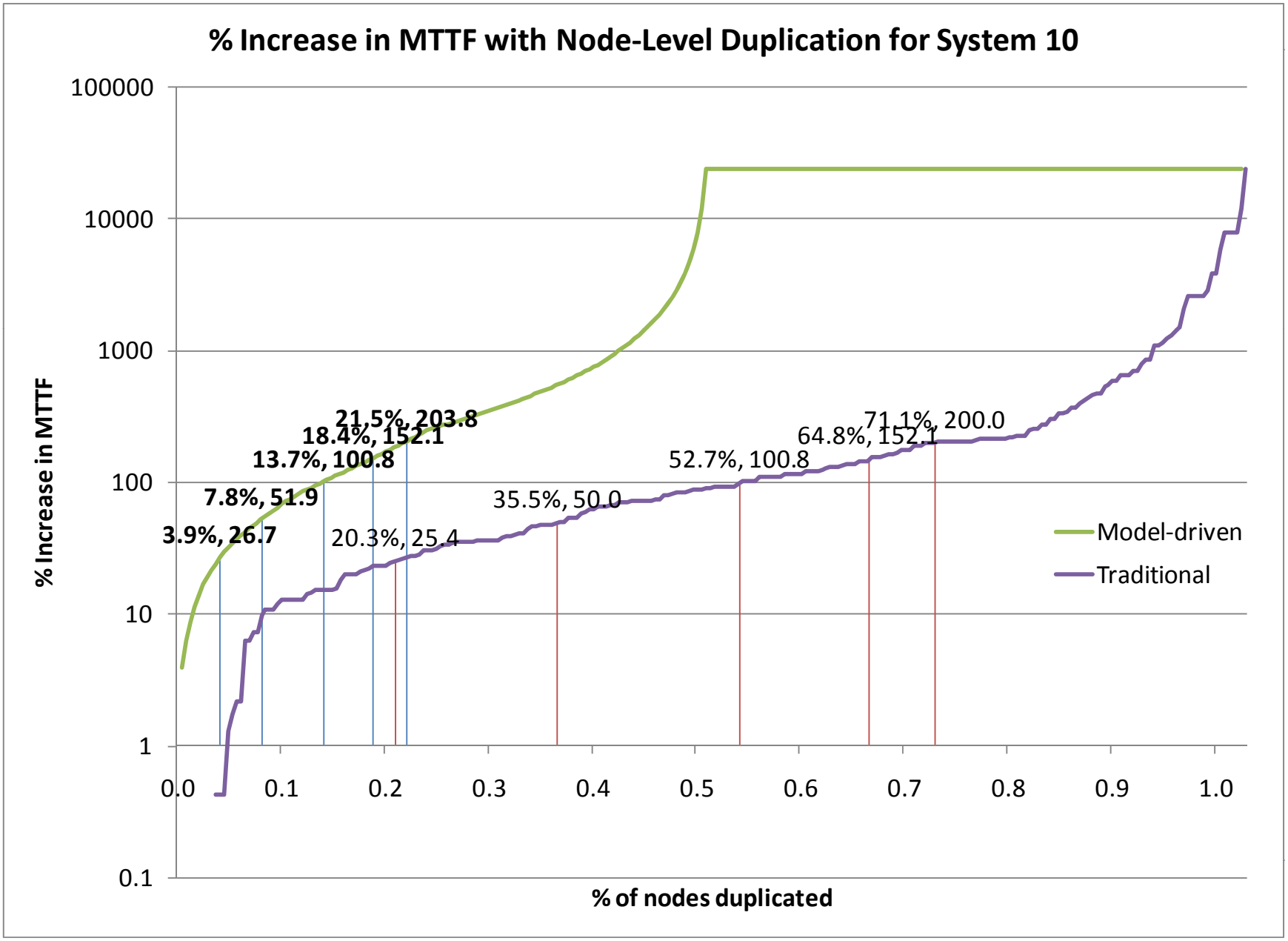
Incremental duplication

- Given a target MTTF for a system nodes are selected incrementally for duplication till the target MTTF is reached
- With no information of node failure rate
  - Assume all nodes fail uniformly (Traditional Approach)
    1. Randomly select a node
    2. Duplicate the node
    3. Recalculate system MTTF
    4. If System MTTF less than Target MTTF go to 1

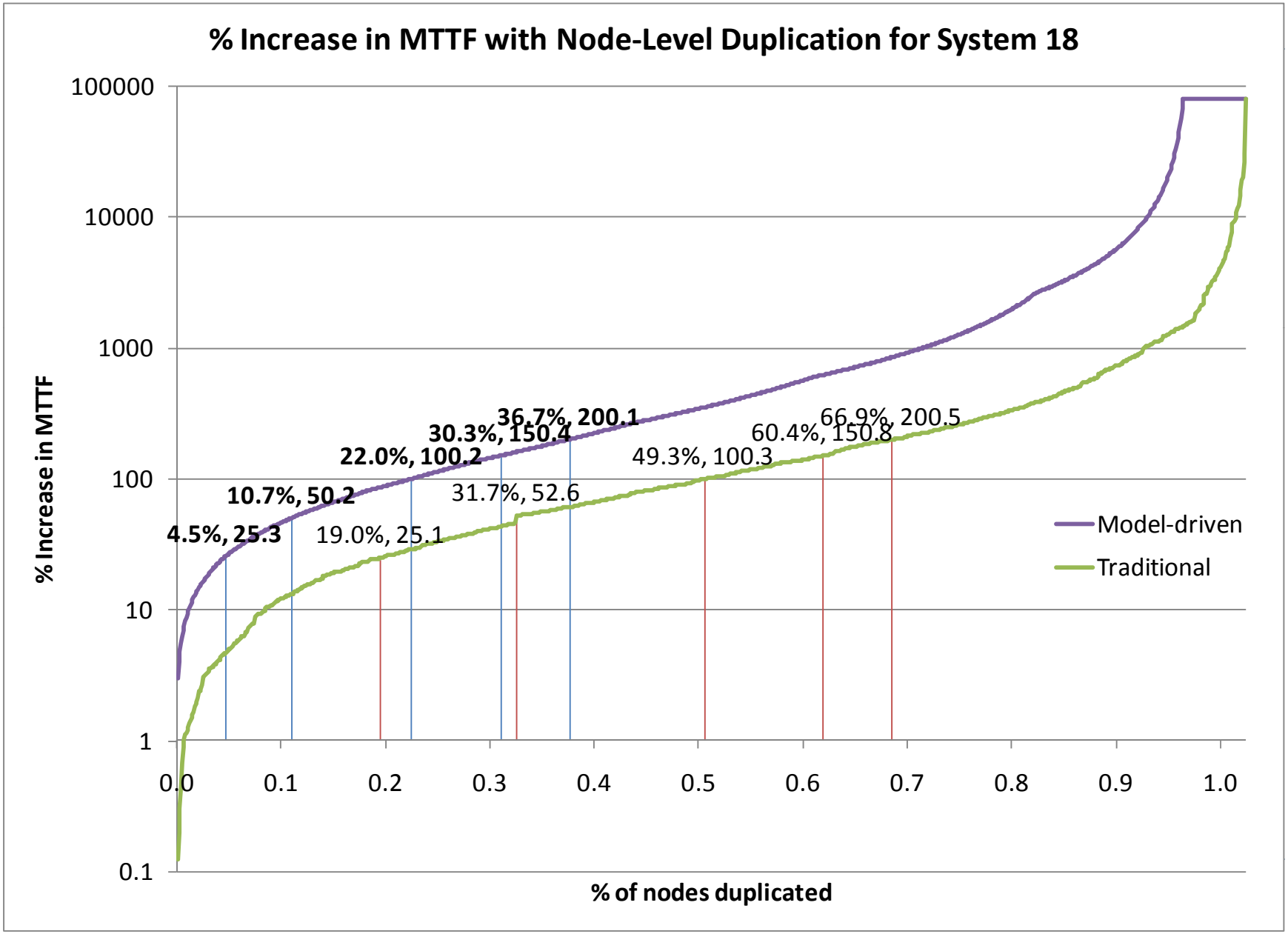
Model-driven node-selection

- Based on the failure rate analysis
  - A model is derived for the coverage provided by duplication
  - Nodes are ordered based on coverage provided by duplicating them
- Model driven approach
  1. Nodes are selected based on the above ordering
  2. Duplicate the node
  3. Recalculate system MTTF
  4. If System MTTF less than Target MTTF go to 1

Model-driven vs. Traditional



Model-driven vs. Traditional

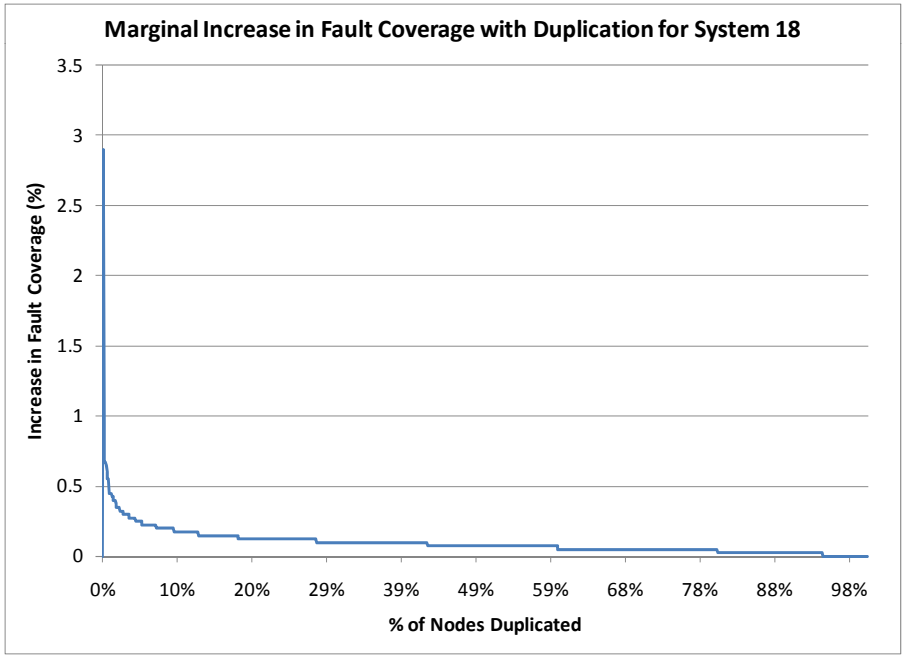


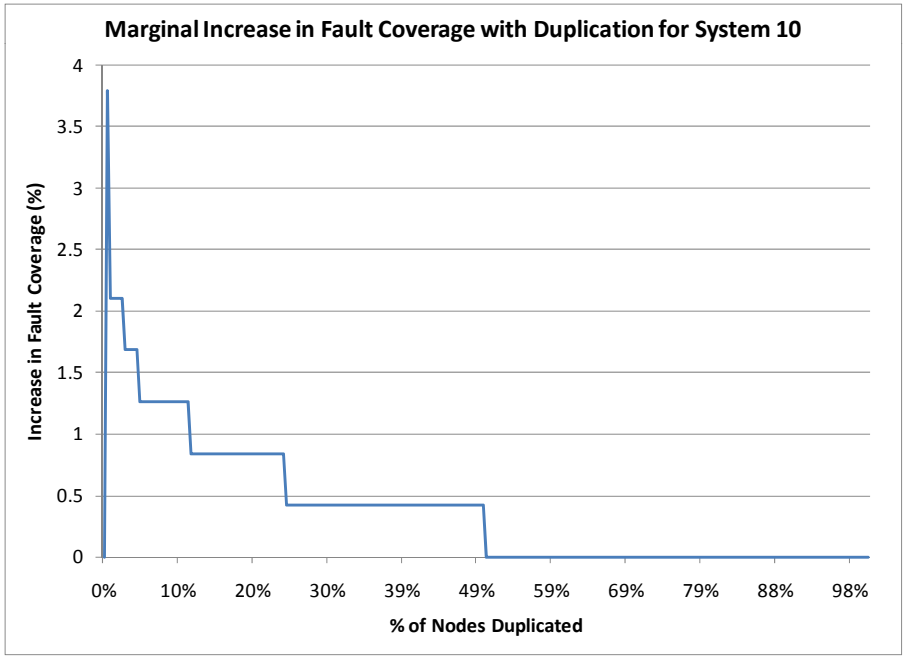
Improvement in #nodes duplicated

Improvement in nodes duplicated		System Number						Average
		10	11	12	13	18	19	
%MTTF Improvement	25	420%	411%	**1700%**	657%	324%	531%	674%
	50	355%	388%	1147%	431%	195%	271%	465%
	100	286%	277%	835%	307%	124%	158%	331%
	150	253%	228%	741%	268%	100%	118%	285%
	200	231%	202%	656%	223%	**82%**	95%	248%
Average		309%	301%	1016%	377%	165%	235%	

Observations

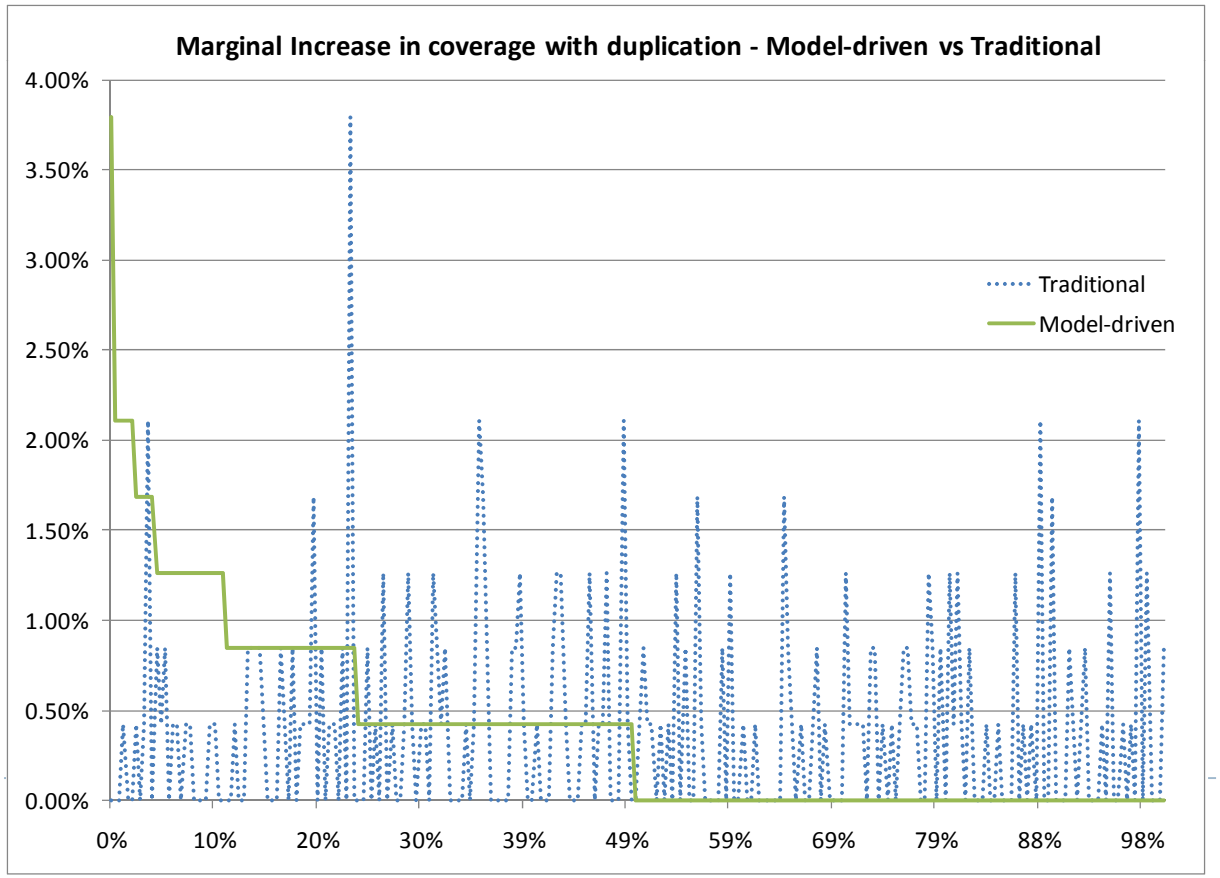
- Avg improvement in # duplicated nodes varies widely between different systems
  - Due to the variation in the failure distribution





Observations

- Improvement achieved in #duplicated nodes decreases for higher increments in MTTF
  - Model driven approach essentially greedy – "attempts to pick node with the next best improvement in MTTF"



Summary and conclusions

- Failure behavior of large scale systems was studied using the failure logs
- Arrived at an ordering of nodes such that a target MTTF is achieved after duplicating the least number of nodes selected in that order
- Network configuration and workload have a close correlation with failure behavior
- As compared to random selection of nodes, model-driven approach provides improvements ranging from 82% to 1700%
  - Improvements depend on the target increase in MTTF and the failure distribution of the system