
Update: Failure data collection & analysis

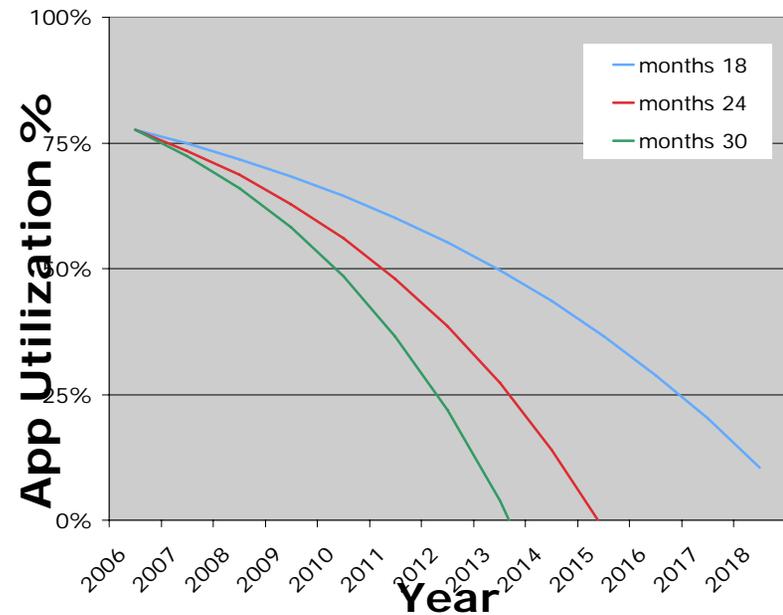
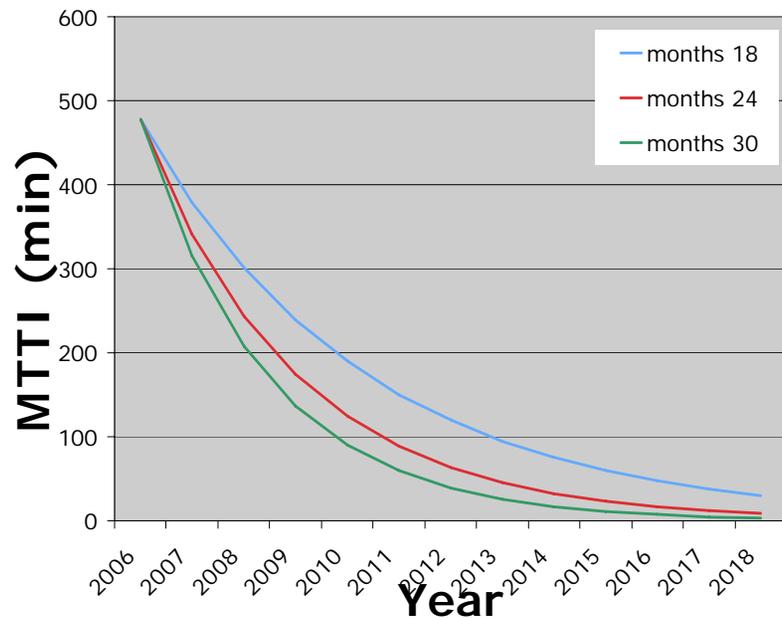
Bianca Schroeder
Joint with Garth Gibson
Carnegie Mellon University

SciDAC Petascale Data Storage Institute (PDSI)

www.pdsi-scidac.org

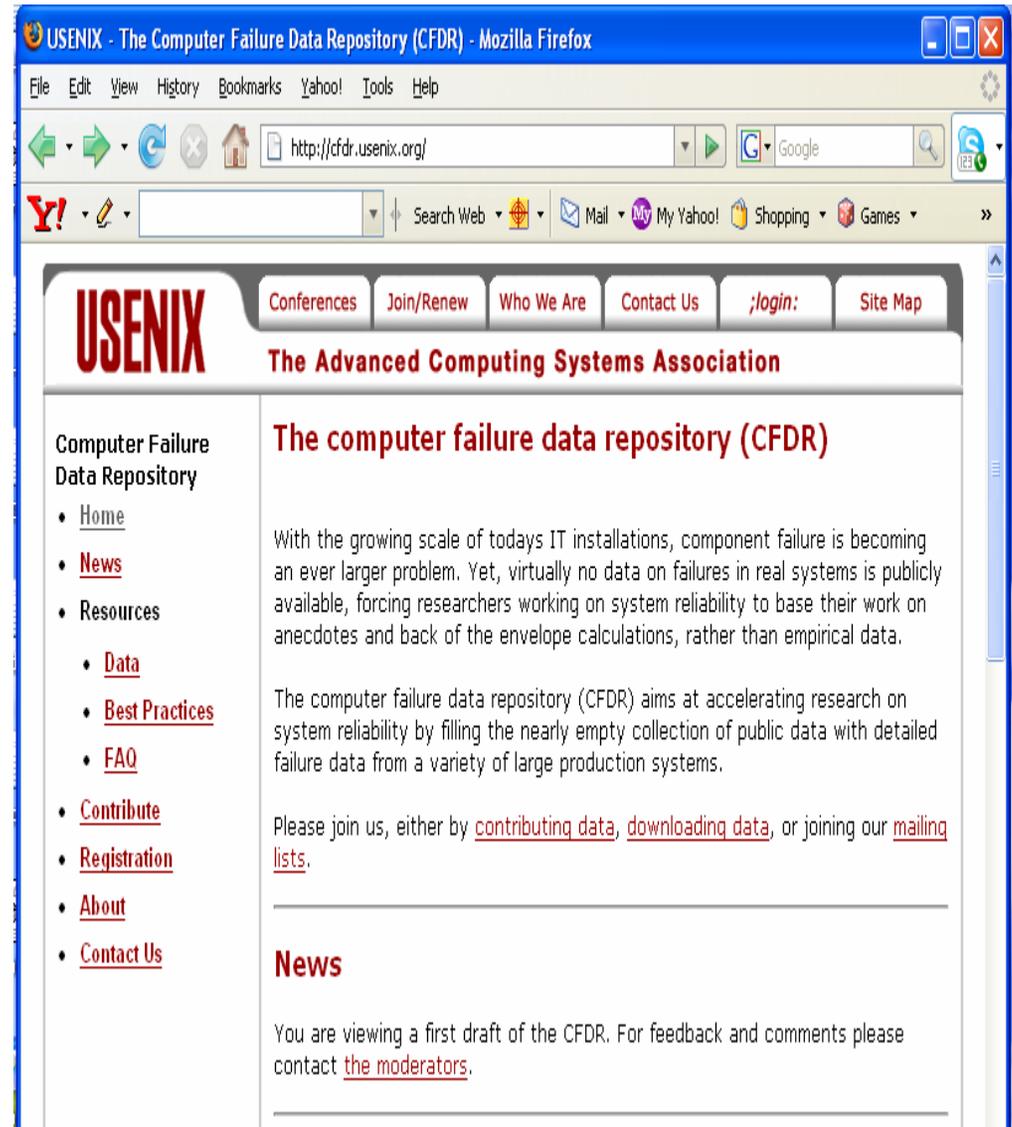
Motivation

- Petascale computing is coming
 - Orders of magnitude more components
 - **Orders of magnitude more failures**
- **Need raw data for better understanding of failures**



The computer failure data repository (CFDR)

- Gather & publish real failure data
- Community effort
 - Usenix clearinghouse
- Data on all aspects of system failure
- Anonymized as needed



The screenshot shows the website for the Computer Failure Data Repository (CFDR) hosted by USENIX. The browser window title is "USENIX - The Computer Failure Data Repository (CFDR) - Mozilla Firefox". The address bar shows "http://cfdr.usenix.org/". The website features a navigation menu with links for "Conferences", "Join/Renew", "Who We Are", "Contact Us", "login:", and "Site Map". The main content area is titled "The computer failure data repository (CFDR)" and includes a paragraph explaining the repository's purpose: "With the growing scale of today's IT installations, component failure is becoming an ever larger problem. Yet, virtually no data on failures in real systems is publicly available, forcing researchers working on system reliability to base their work on anecdotes and back of the envelope calculations, rather than empirical data. The computer failure data repository (CFDR) aims at accelerating research on system reliability by filling the nearly empty collection of public data with detailed failure data from a variety of large production systems." Below this, there is a call to action: "Please join us, either by [contributing data](#), [downloading data](#), or joining our [mailing lists](#)." A "News" section at the bottom states: "You are viewing a first draft of the CFDR. For feedback and comments please contact [the moderators](#)."

Available data

- Downloaded 900 times in 6 months
- Used in at least 3 SC'07 papers
- Please send us pointers!

9 years of node outages
[DSN'06, TDSC]
[SciDAC'07]

Error logs
[DSN'07]

I/O specific failures

Name	Time Period	System	Description
LANL	Dec 96 - Nov 05	HPC clusters	The data covers node outages at 22 cluster systems at LANL , including a total of 4,750 nodes and 24,101 processors. Some job logs and error logs are available as well.
HPC1	Aug 01 - May 06	HPC cluster	The data covers hardware replacements at a 765 node cluster with more than 3,000 hard drives.
HPC2	Jan 04 - Jul 06	HPC cluster	Hard drive replacements in a 256 node cluster with 520 drives.
HPC3	Dec 05 - Nov 06	HPC cluster	Hard drive replacements observed in a 1,532-node HPC cluster with more than 14,000 drives.
HPC4	2004 - 2006	HPC cluster	Error logs collected at 5 supercomputing systems at SNL and LLNL , ranging from 512 to 131072 processors.
PNNL	Nov 03 - Sep 07	HPC cluster	Hardware failures recorded on the MPP2 system (a 980 node HPC cluster) at PNNL .
NERSC	2001 - 2006	HPC cluster	I/O specific failures collected at a number of production systems at NERSC .

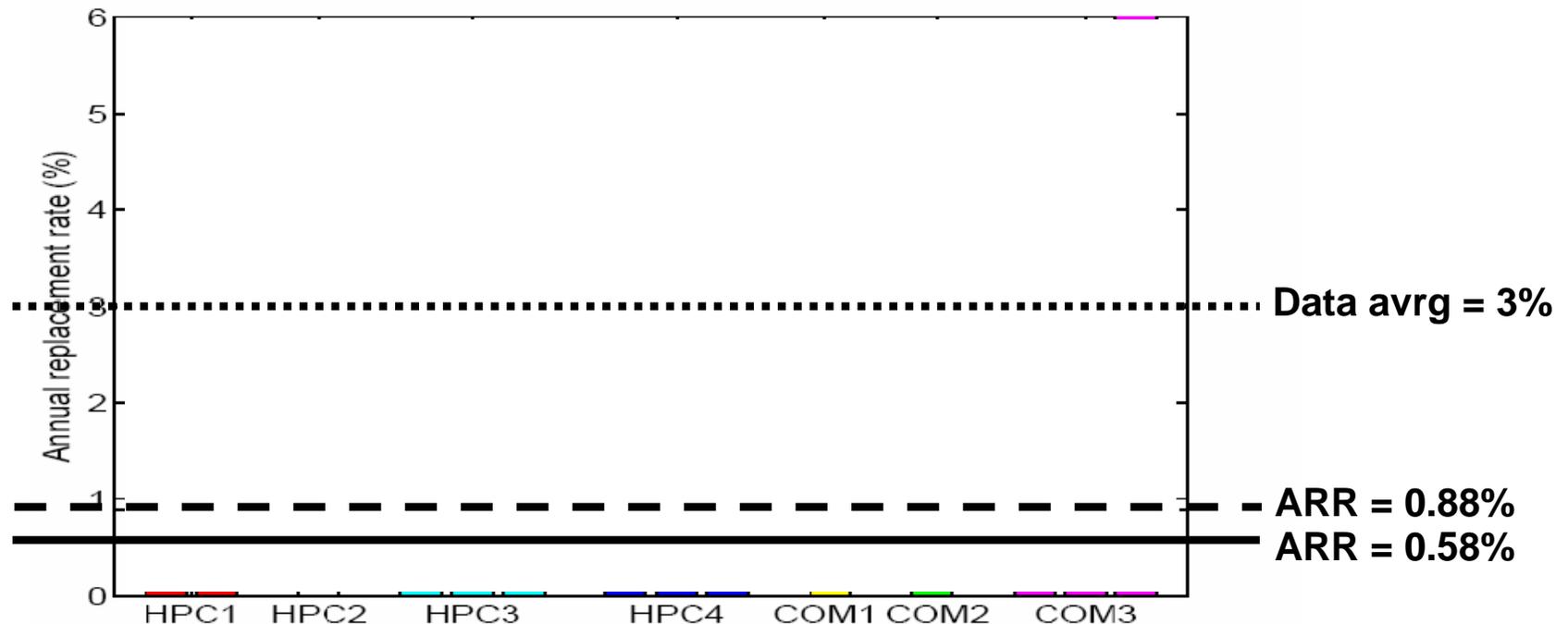
Hardware / disk drive failures
[FAST'07, TOS]

Data not available (yet):

- [FAST'07 Google] study of hard drive replacements
- [Sigmetrics'07 NetApp] study of media errors

How often do drives really fail?

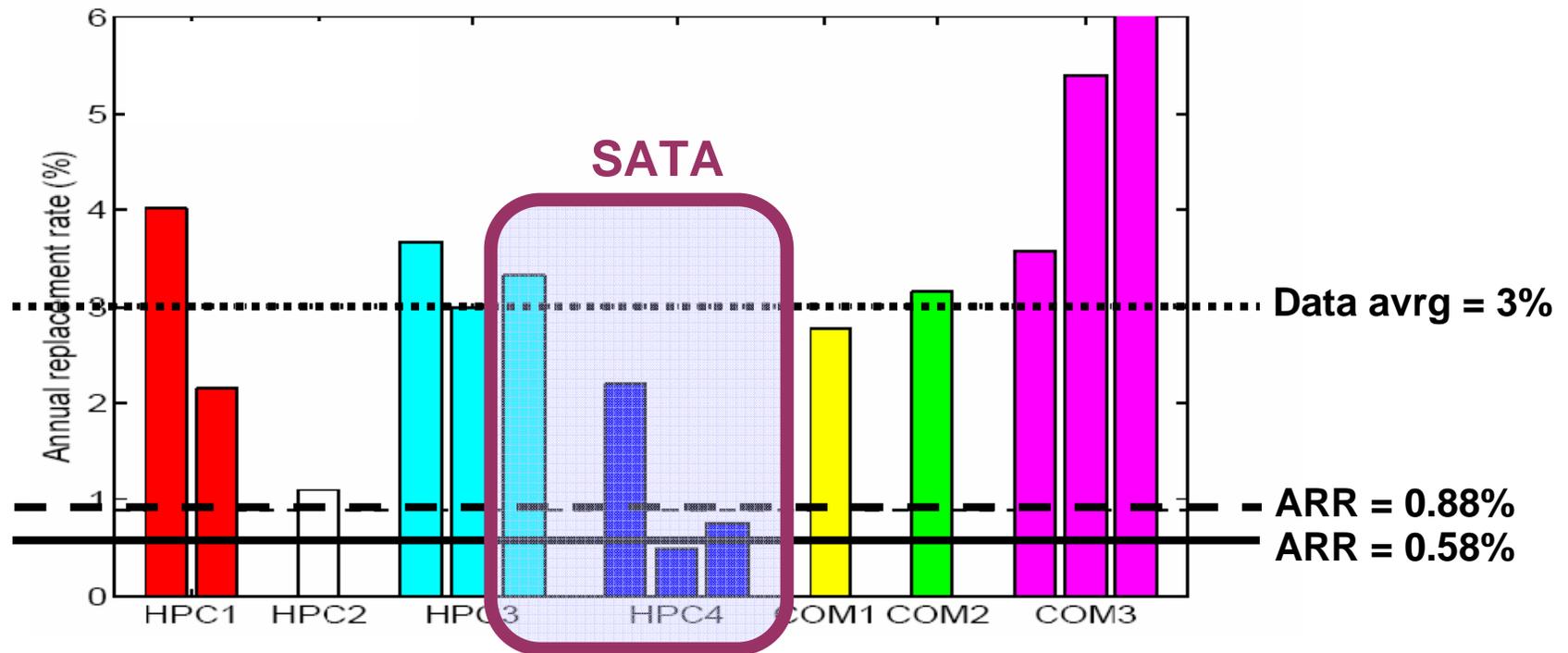
- Vendor datasheets: Annual replacement rates (ARR) of 0.58 - 0.88 %



- Field replacement rates are significantly higher than what vendor datasheets suggest

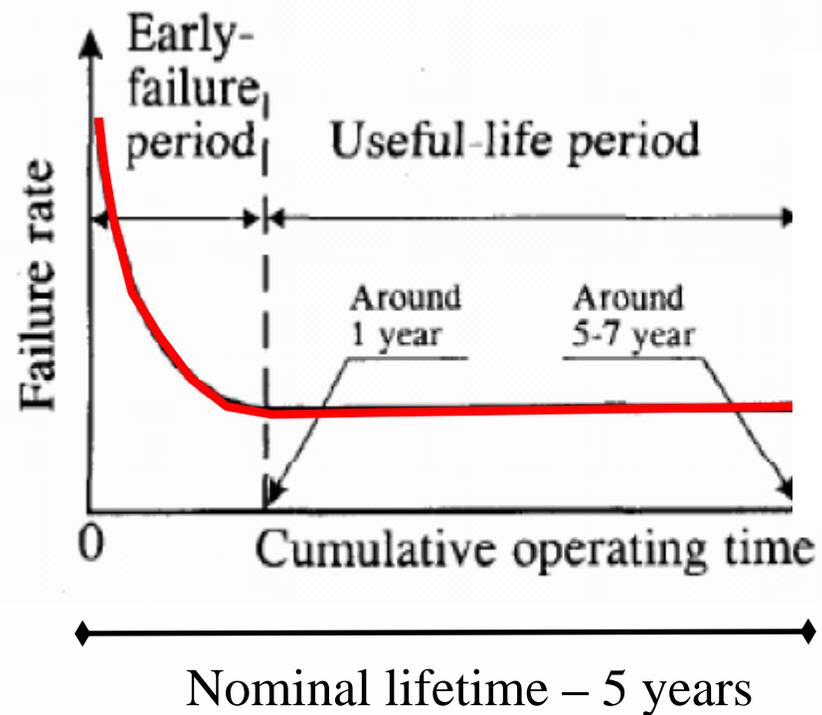
How often do drives really fail?

- Vendor datasheets: Annual replacement rates (ARR) of 0.58 - 0.88 %

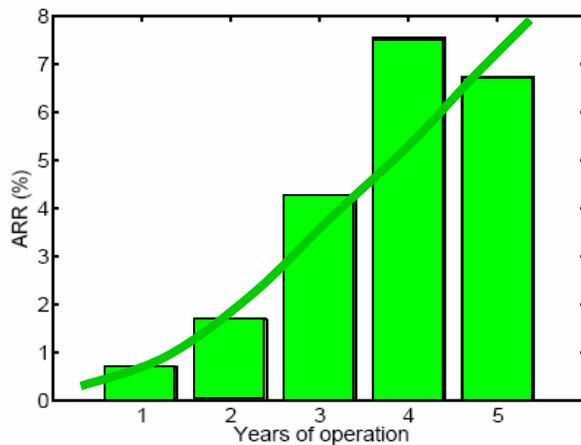
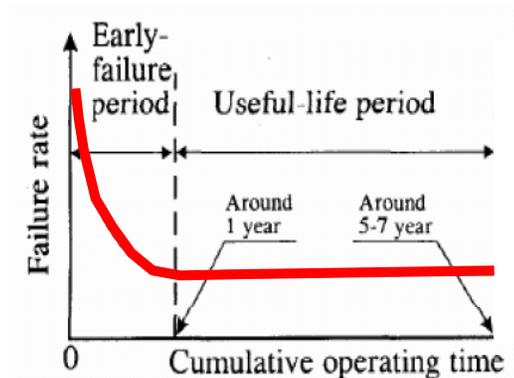


- No evidence that SATA disks exhibit higher replacement rates than SCSI or FC disks

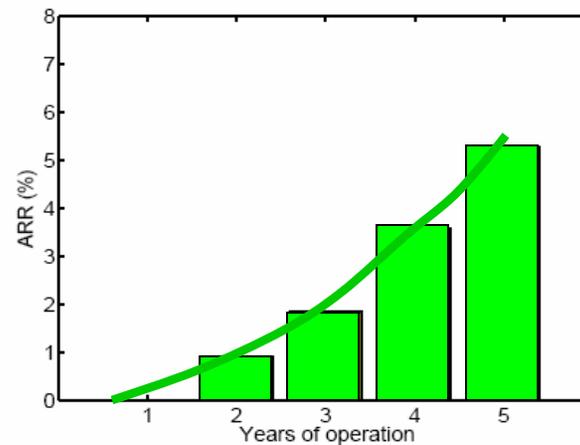
Replacement rate as a function of age - model



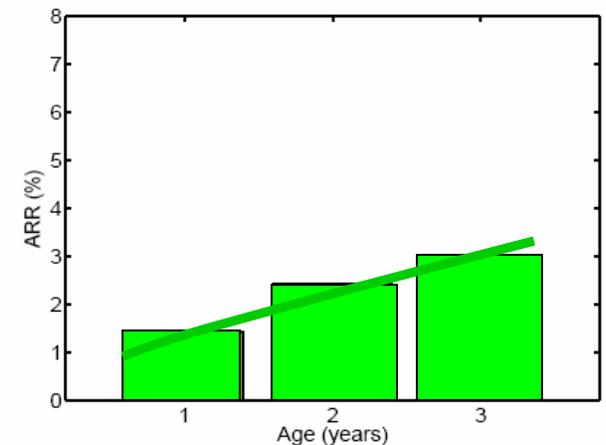
Replacement rate as a function of age



HPC1 (compute nodes)



HPC1 (filesystem nodes)

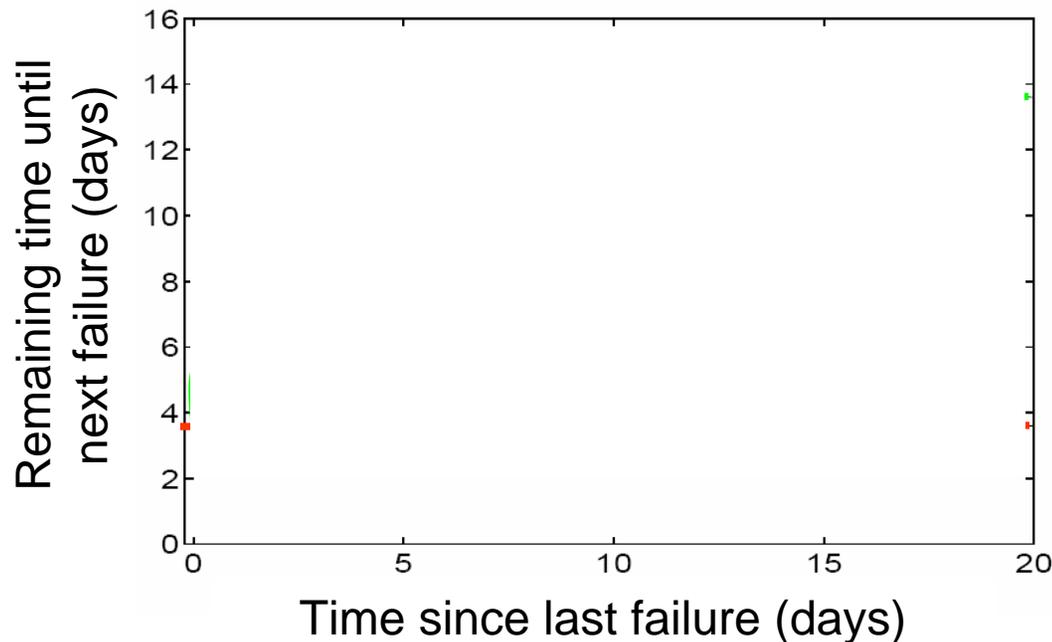


HPC4

- Wear-out seems to set in earlier than often assumed
- Infant mortality not significant

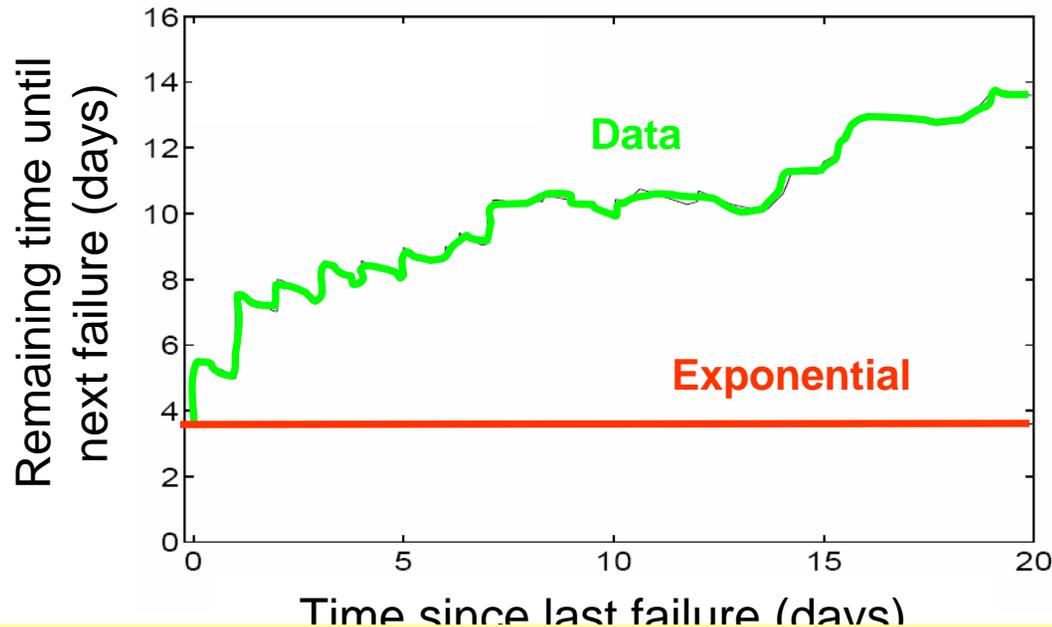
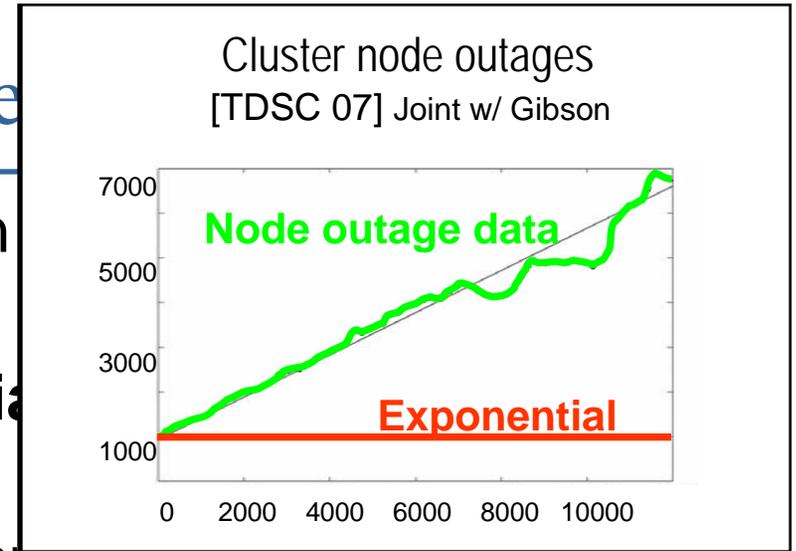
Statistical properties of time between failure?

- *Common assumption:* Time between failure follows an exponential distribution
- Real data does not follow **exponential** distribution
 - Variability is higher ($C^2 = 2.5 - 12$)
 - Weibull distribution with shape parameter $s < 1$ is better fit



Statistical properties of time

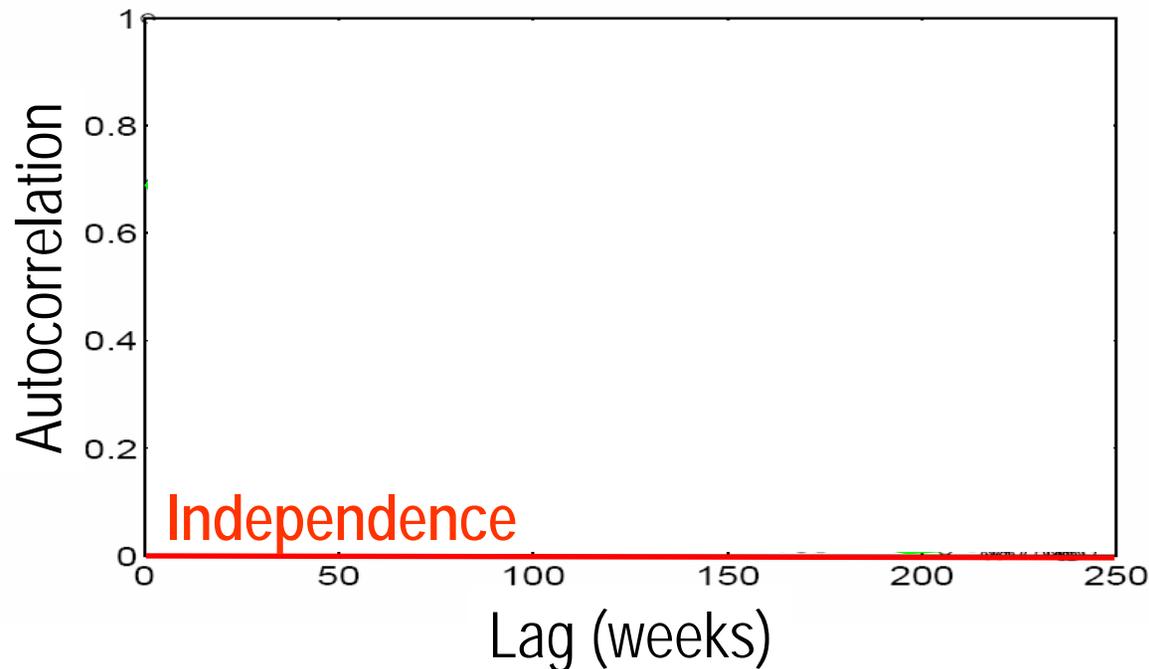
- *Common assumption:* Time between exponential distribution
- Real data does not follow **exponential**
 - Variability is higher ($C^2 = 2.5 - 12$)
 - Weibull distribution with shape parameter $s < 1$ is better fit



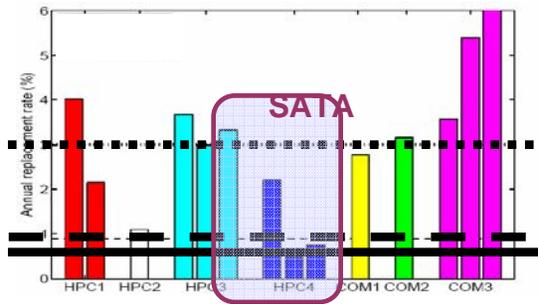
- First published data that allows rejection of exponential assumption for time between drive failures

Statistical properties of time between failure

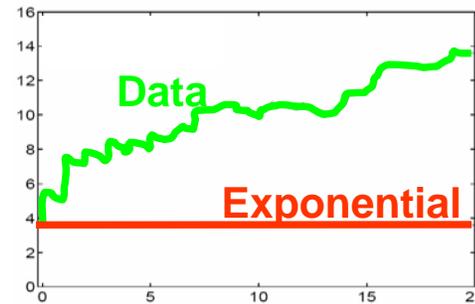
- *Common assumption:* Failures are independent
- Real data shows correlations at various levels including
 - auto-correlation
 - long-range dependence.



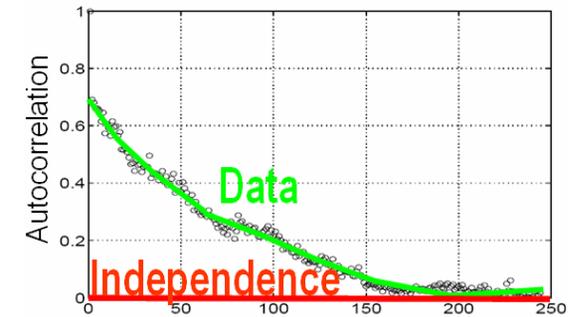
Many common assumptions not realistic



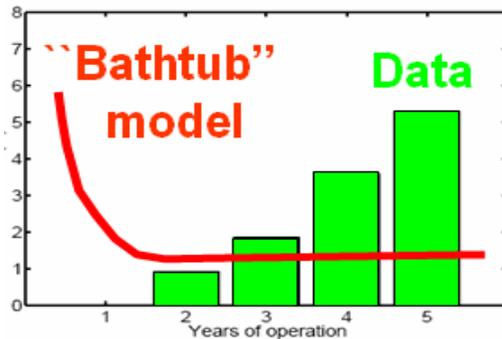
• Repl. rates higher than specs



• Time between failure not exponential



• Failures not independent



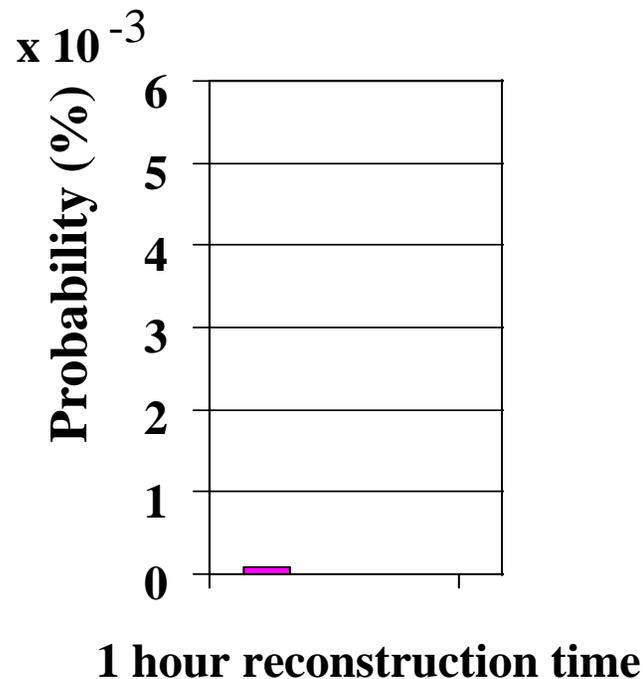
• "Bathtub" model not realistic

! Important to work with real data! !

Estimating probability of data loss in RAID

- Depends on probability of second failure during reconstruction

■ Standard approach: Use datasheet MTTF and exponential distribution

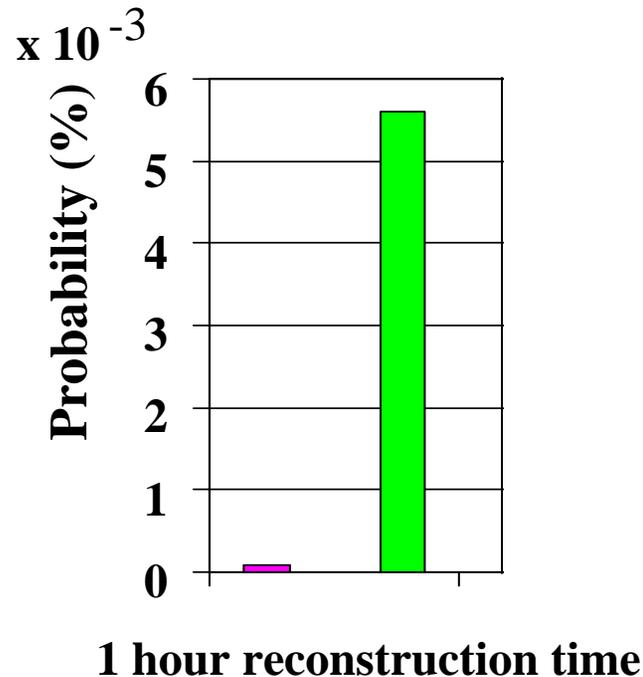


Estimating probability of data loss in RAID

- Depends on probability of second failure during reconstruction

 Standard approach: Use datasheet MTTF and exponential distribution

 Estimate based on data



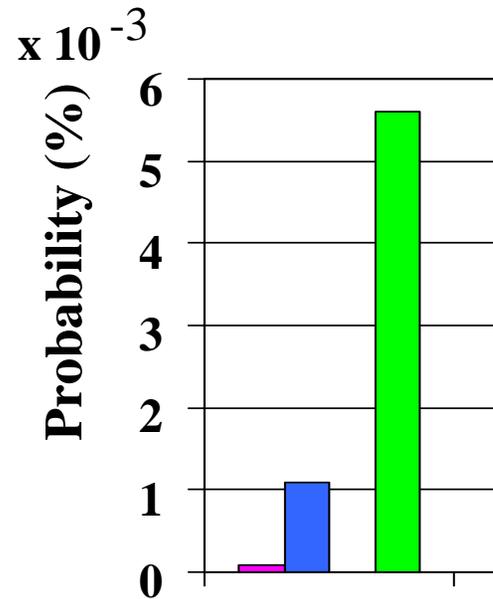
Estimating probability of data loss in RAID

- Depends on probability of second failure during reconstruction

 Standard approach: Use datasheet MTTF and exponential distribution

 Use measured MTTF and exponential distribution

 Estimate based on data

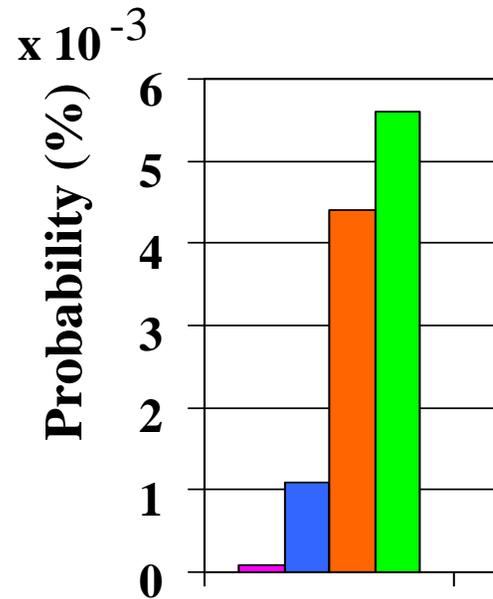


1 hour reconstruction time

Estimating probability of data loss in RAID

- Depends on probability of second failure during reconstruction

- Standard approach: Use datasheet MTTF and exponential distribution
- Use measured MTTF and exponential distribution
- Use measured MTTF and Weibull distribution
- Estimate based on data

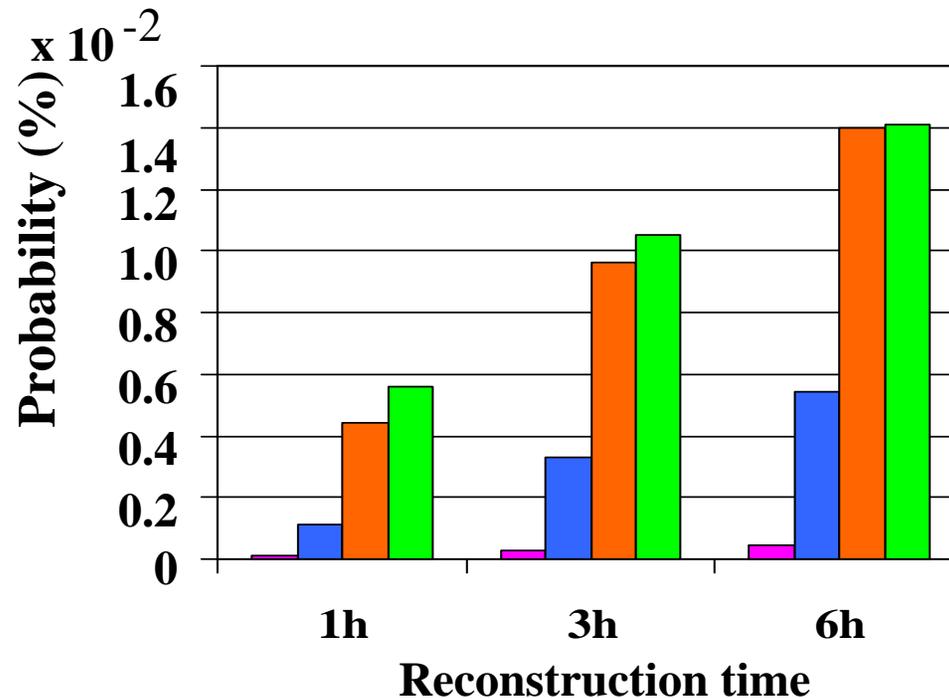


1 hour reconstruction time

Estimating probability of data loss in RAID

- Depends on probability of second failure during reconstruction

- Standard approach: Use datasheet MTTF and exponential distribution
- Use measured MTTF and exponential distribution
- Use measured MTTF and Weibull distribution
- Estimate based on data



Conclusion

- Many challenges in Petascale reliability ahead
- Failures don't always look as expected
- Sharing failure data powerful for systems research
- **Need to continue to collect & publish more data!**
- **THANKS to those who have contributed data!!!**

Do you have any data
to contribute?

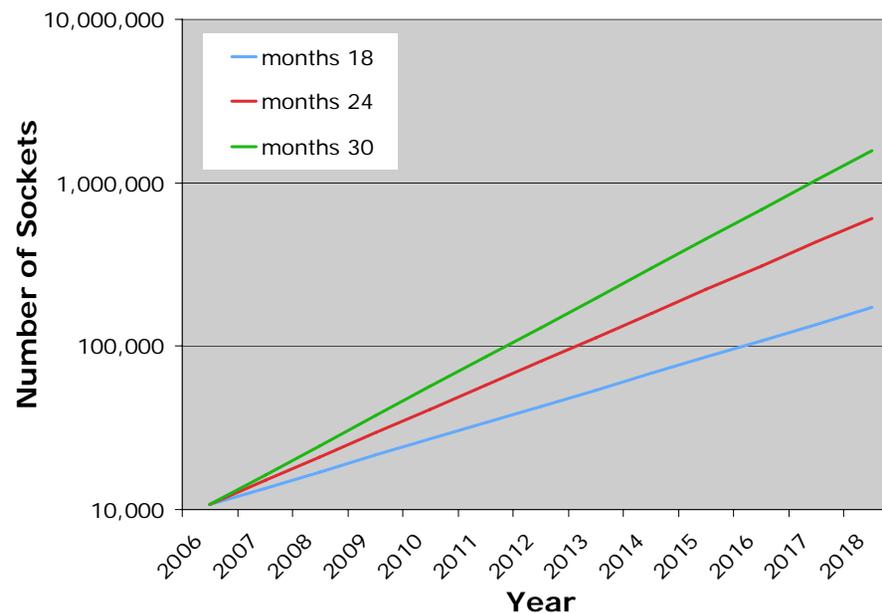
Contact us:
{bianca,garth}@cs.cmu.edu

Thanks!
Questions?

Backup slides

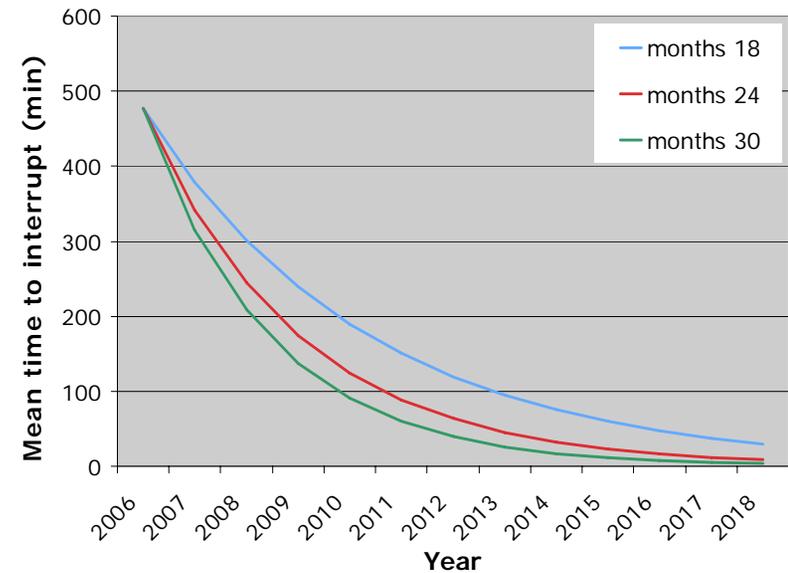
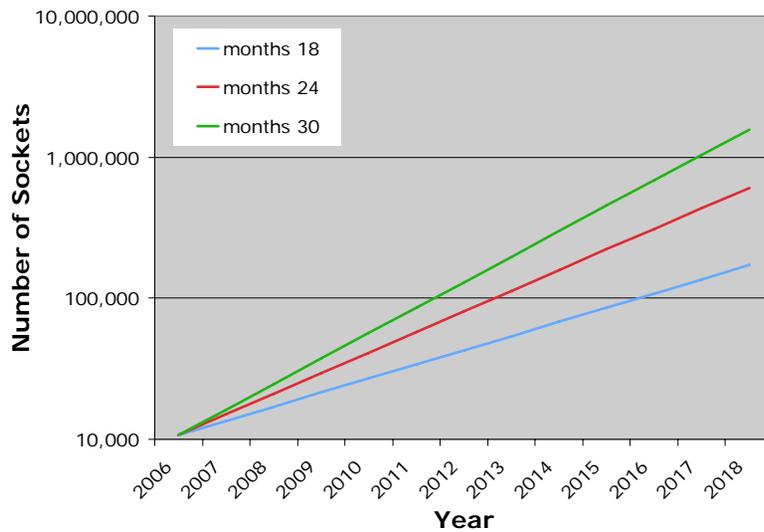
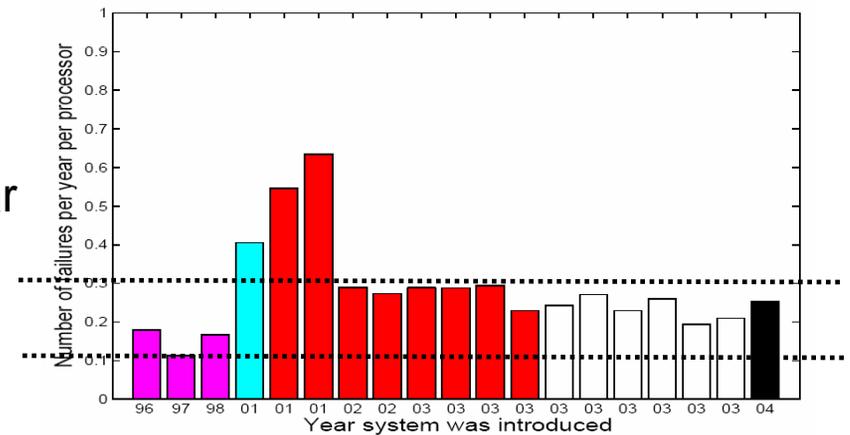
Petascale projections

- Continued top500.org annual 2X peak FLOPS
 - Set to 1 PF plan for ORNL Baker, LANL Roadrunner in 2008
- Cycle time flat; Cores/chip on Moore's law
 - Consider 2X cores per chip every 18, 24, 30 months



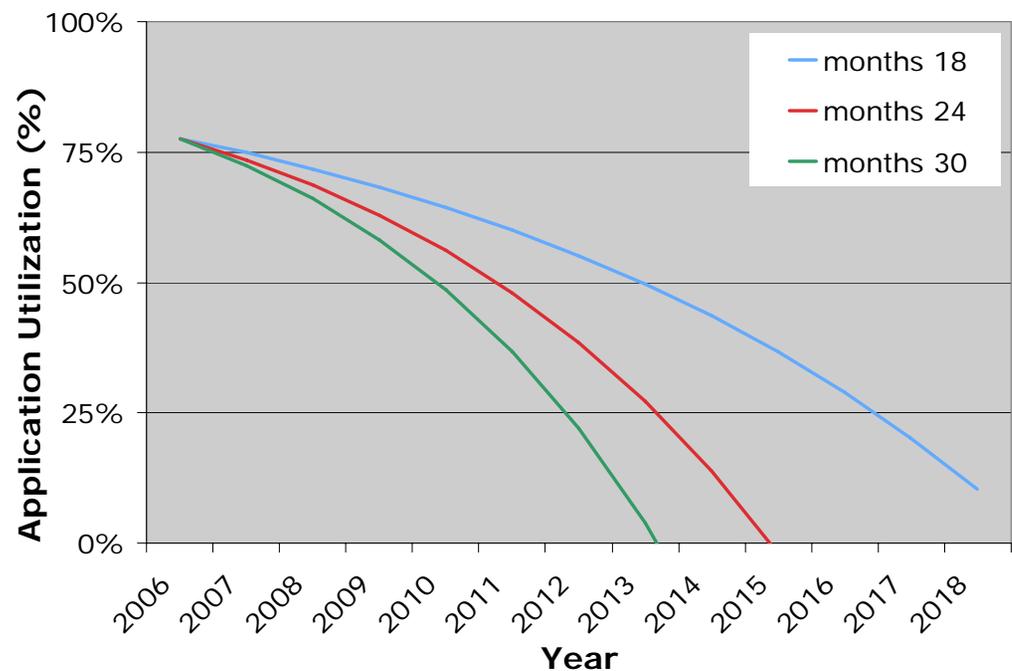
Petascale projections: future MTTIs

- Failure rate grows with number of chips
 - Stable over time
 - Assume optimistic 0.1 failures per year per socket (vs. historic 0.25)



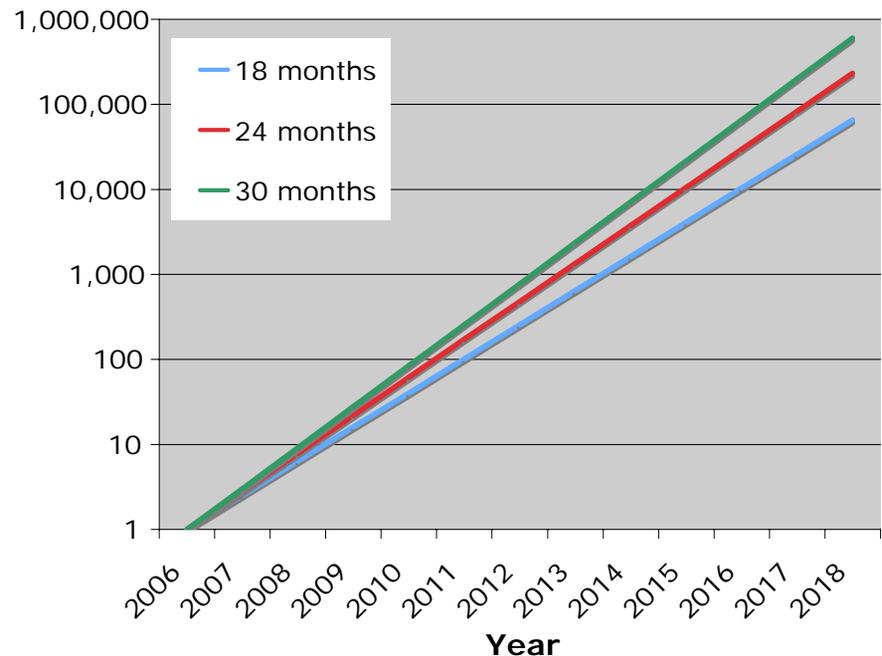
Petascale projections: app's utilization

- Periodic (p) app pause to capture checkpoint (t)
- On failure, roll back & restart from checkpoint
- Balanced: Mem, disk speed track FLOPS (constant t)
 - $1 - \text{App util} = t / p + p / (2 * \text{MTTI}); p^2 = 2 * t * \text{MTTI}$
 - If MTTI was constant, app utilization would be too
- But MTTI drops
- So Application utilization drops
- Half machine gone soon
- Not acceptable



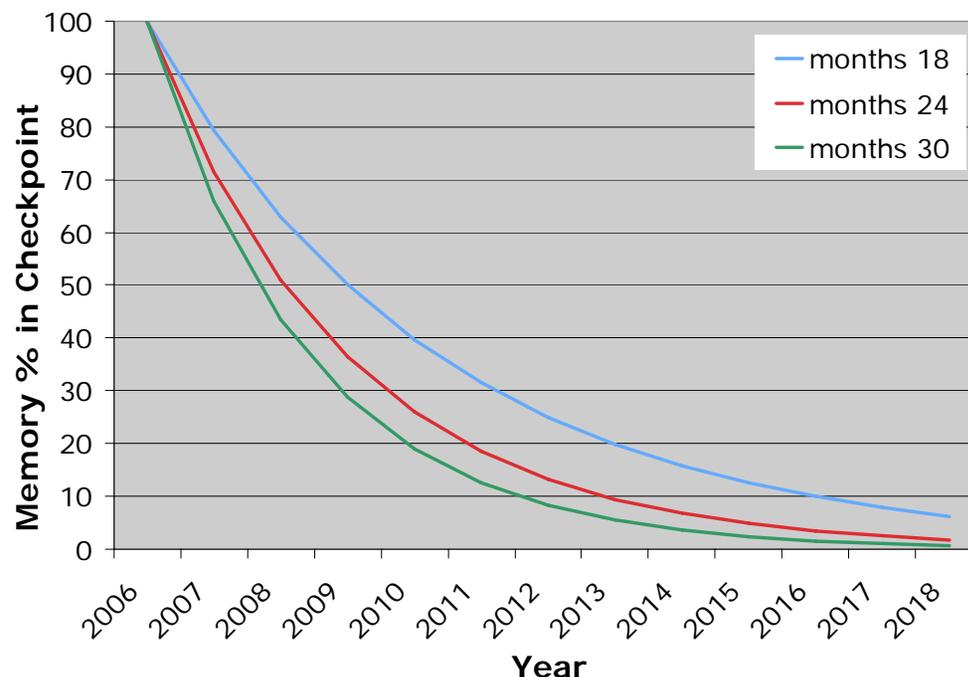
Storage bandwidth to the rescue?

- Increase storage bandwidth to counter for MTTI?
- First, balance means storage bandwidth tracks FLOPS, 2X per year, but disks 20% faster each year
 - Number of disks up 67% each year just for balance
- Doesn't counter MTTI
 - # Disks up 130% / year !
 - Faster than sockets, faster than FLOPS!
 - If system cost grows as # disks vs # sockets
 - Total costs increasingly going into storage (even just for balance)



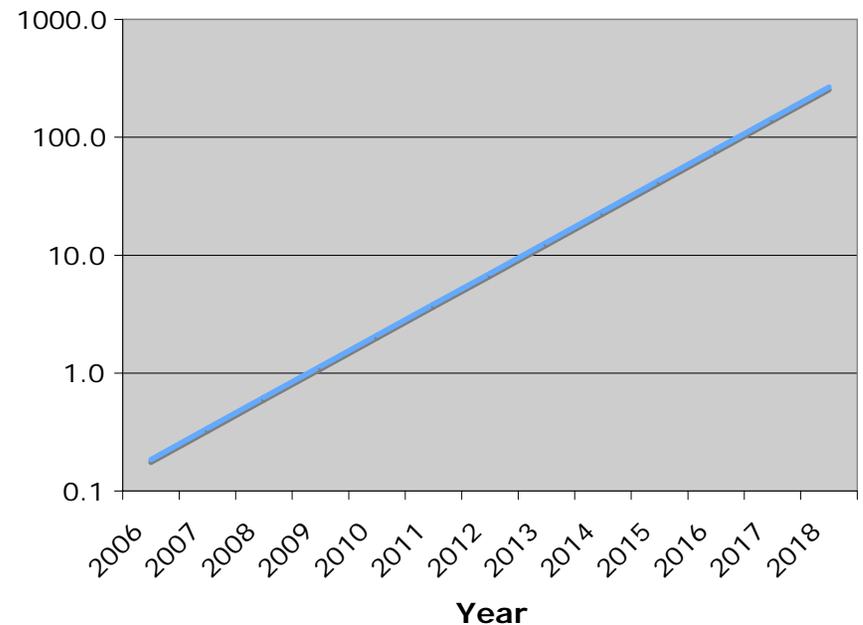
Applications squeeze checkpoints?

- So far, assumed checkpoint size is memory
- Could Apps counter MTTI with compression?
- Size of checkpoint has to decrease with MTTI
 - Smaller fraction of memory with each machine
 - Drop 25-50% per year
- Soon only 50% memory in checkpoint ...



While on storage issues ...

- Increasing disk bandwidth: more disks & disk failures
 - Data shows 3% per year are replaced
- RAID (level 5, 6 or stronger codes) protect data
 - At cost of online reconstruction of all lost data
 - Larger disks: longer reconstructions, hours become days
- Consider # concurrent reconstructions
- 10-20% now, but
- Soon 100s of concurrent reconstructions
- Storage does not have checkpoint/restart model
- Design normal case for many failures



Smaller applications escape

- X

Change fault tolerance scheme?

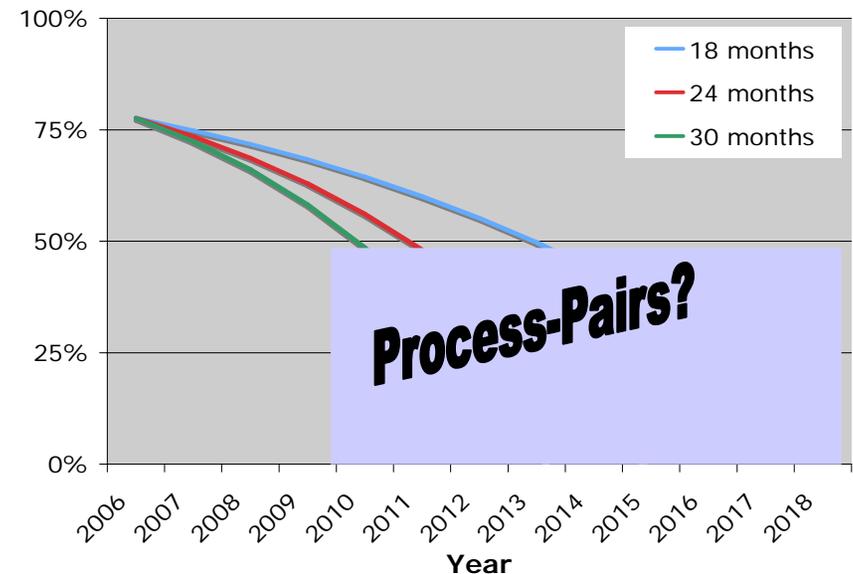
- Classic reliable computing: process-pairs
 - Distributed, parallel simulation as transaction (message) processing
 - Automation possible w/ hypervisors
- Deliver all incoming messages to both
- Match outgoing messages from both
- 50% hardware overhead + slowdown from synch
- But if App Utilization is falling under 50% anyway
- No stopping to checkpoint
 - Less pressure on storage bandwidth except for visualization checkpoints

A NonStop* Kernel

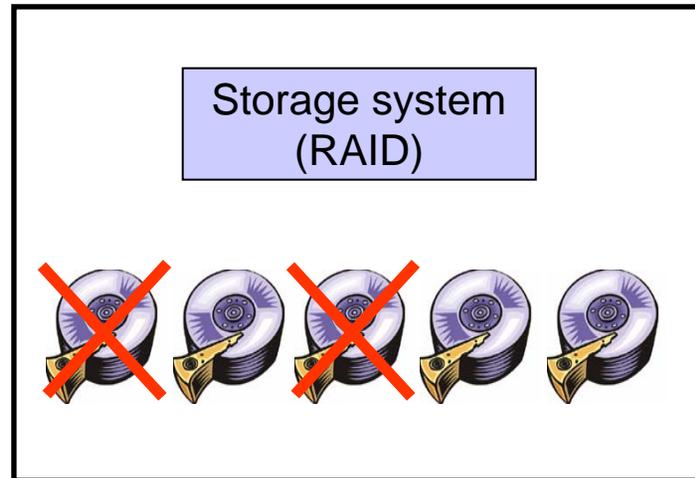
Joel F. Bartlett
Tandem Computers Inc.

Abstract © 1981 ACM 0-89791-062-1-12/81-0022

The Tandem NonStop System is a fault-tolerant [1], expandable, and distributed computer system designed expressly for online transaction processing. This paper describes the key primitives of the kernel of the operating system. The first section describes the basic hardware building blocks and introduces their software analogs: processes and messages. Using these primitives, a mechanism that allows fault-tolerant resource access, the process-pair, is described. The paper concludes with some observations on this type of system structure and on actual use of the system.



Probability of losing data in a RAID?

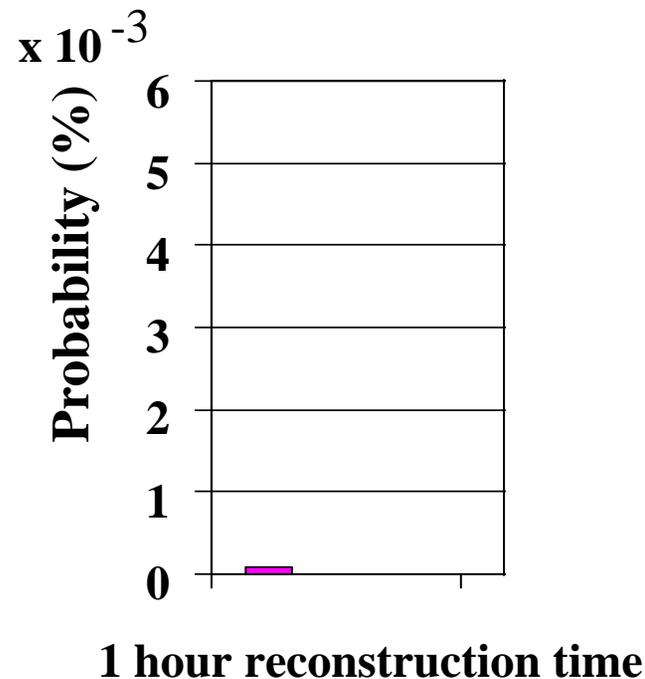


- Depends on probability that after one drive fails, a second drive fails while reconstructing data.

Estimating probability of data loss

- Depends on probability of second failure during reconstruction

■ Standard approach: Use datasheet MTTF and exponential distribution

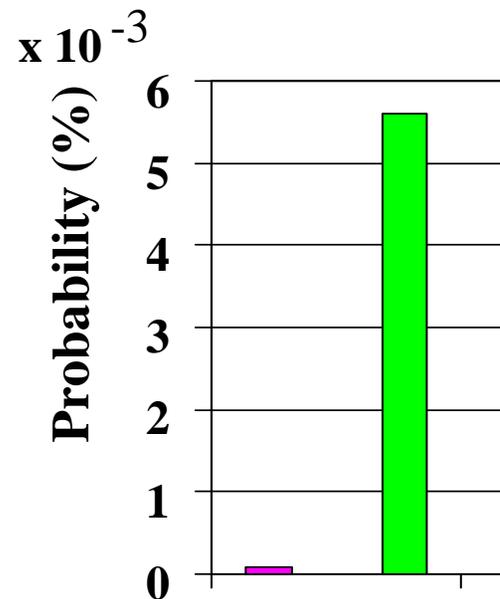


Estimating probability of data loss

- Depends on probability of second failure during reconstruction

 Standard approach: Use datasheet MTTF and exponential distribution

 Estimate based on data



1 hour reconstruction time

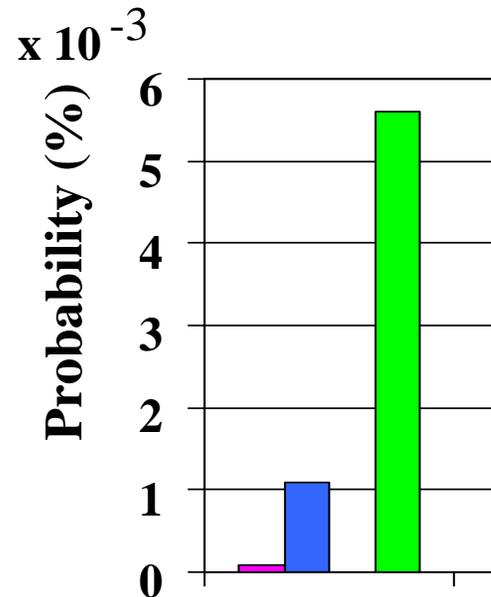
Estimating probability of data loss

- Depends on probability of second failure during reconstruction

 Standard approach: Use datasheet MTTF and exponential distribution

 Use measured MTTF and exponential distribution

 Estimate based on data

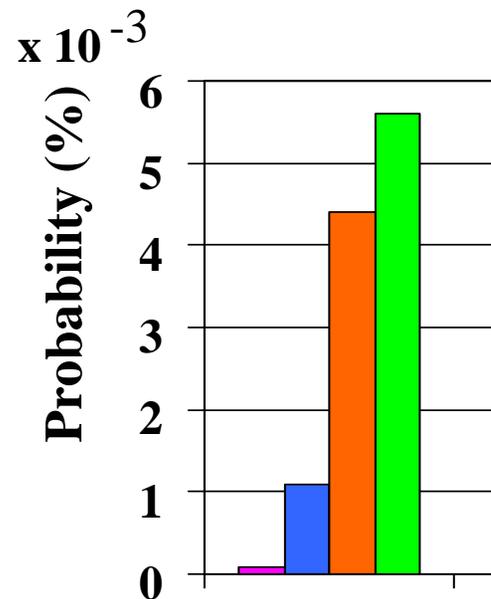


1 hour reconstruction time

Estimating probability of data loss

- Depends on probability of second failure during reconstruction

- Standard approach: Use datasheet MTTF and exponential distribution
- Use measured MTTF and exponential distribution
- Use measured MTTF and Weibull distribution
- Estimate based on data



1 hour reconstruction time