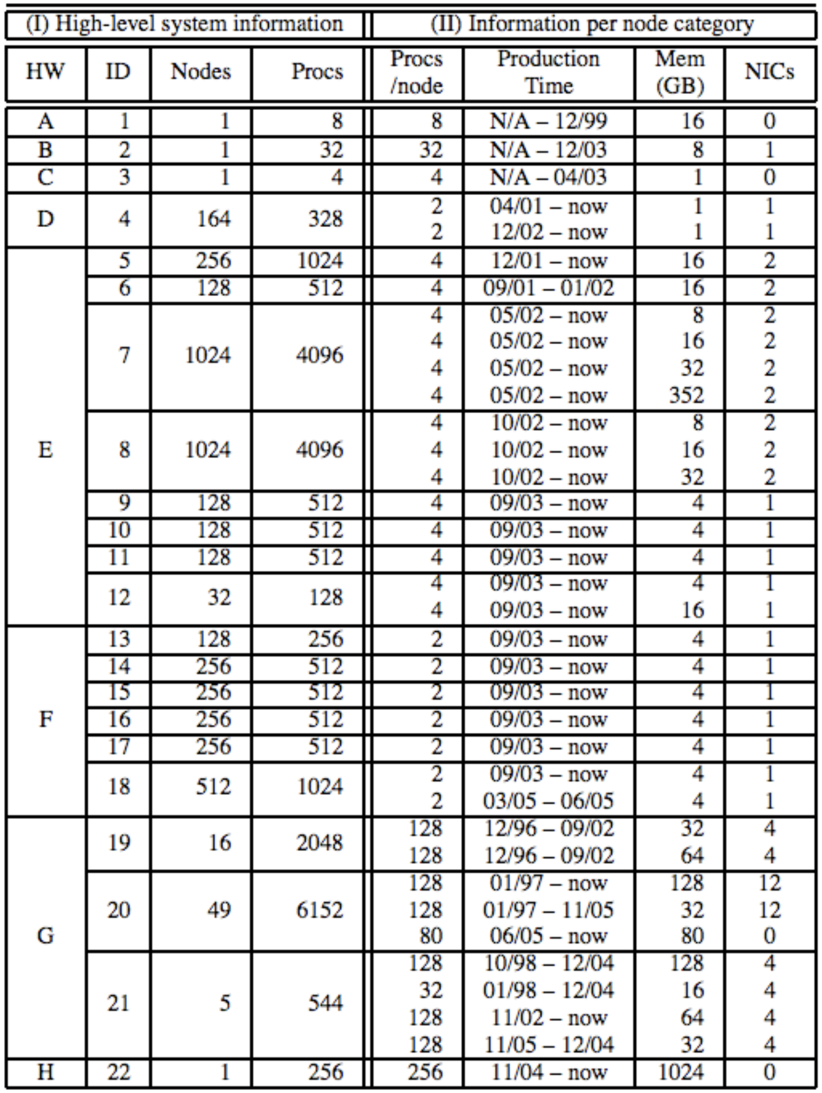
Failure in Supercomputers in the Post-Terascale Era

Thanks to:
Garth Gibson, Carnegie Mellon University and Panasas Inc.
DOE SciDAC Petascale Data Storage Institute (PDSI), www.pdsi-scidac.org
w/ Bianca Schroeder, Carnegie Mellon University (Univ. of Toronto soon)

& w/ Los Alamos (G. Grider), Lawrence Berkeley (W. Kramer), Sandia (L. Ward),
Oak Ridge (P. Roth), and Pacific Northwest (E. Felix) National Laboratories, and
Univ. of California at Santa Cruz (D. Long), and Univ. of Michigan (P. Honeyman)

LANL interrupt history

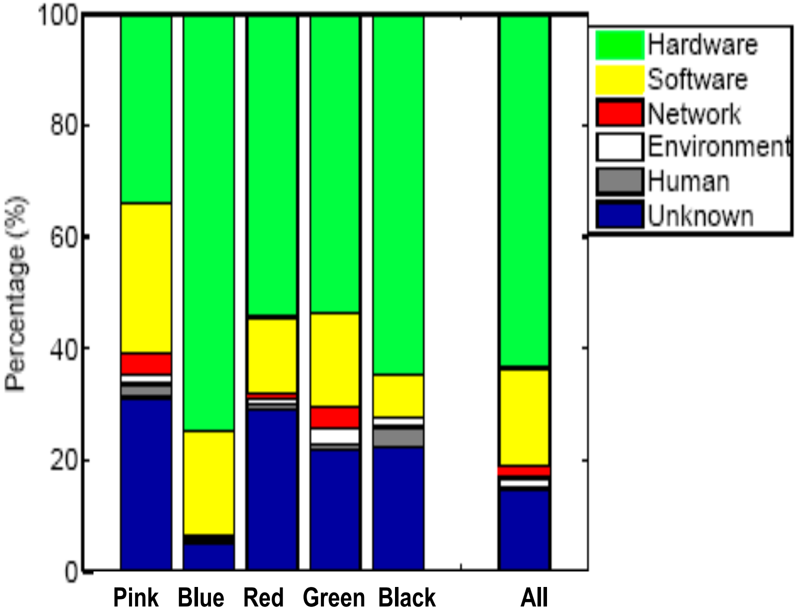
- Los Alamos releases root cause logs for:
    - 23,000 events causing application interruption
    - 22 clusters & 5000 nodes
    - Covers 9 years & continues
- Kicks off our work understanding pressure on storage bandwidth
    - Checkpoint/restart
- More recent failure logs released from NERSC, PNNL, PSC, 2 anonymous



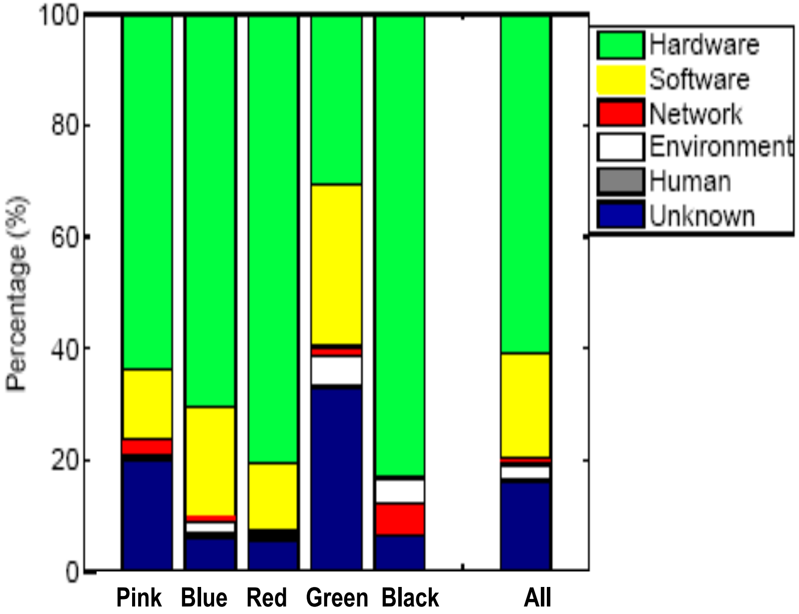
(I) High-level system information				(II) Information per node category			
HW	ID	Nodes	Procs	Procs /node	Production Time	Mem (GB)	NICs
A	1	1	8	8	N/A – 12/99	16	0
B	2	1	32	32	N/A – 12/03	8	1
C	3	1	4	4	N/A – 04/03	1	0
D	4	164	328	2	04/01 – now	1	1
				2	12/02 – now	1	1
E	5	256	1024	4	12/01 – now	16	2
	6	128	512	4	09/01 – 01/02	16	2
	7	1024	4096	4	05/02 – now	8	2
				4	05/02 – now	16	2
				4	05/02 – now	32	2
				4	05/02 – now	352	2
	8	1024	4096	4	10/02 – now	8	2
				4	10/02 – now	16	2
				4	10/02 – now	32	2
	9	128	512	4	09/03 – now	4	1
	10	128	512	4	09/03 – now	4	1
	11	128	512	4	09/03 – now	4	1
	12	32	128	4	09/03 – now	4	1
				4	09/03 – now	16	1
F	13	128	256	2	09/03 – now	4	1
	14	256	512	2	09/03 – now	4	1
	15	256	512	2	09/03 – now	4	1
	16	256	512	2	09/03 – now	4	1
	17	256	512	2	09/03 – now	4	1
	18	512	1024	2	09/03 – now	4	1
				2	03/05 – 06/05	4	1
G	19	16	2048	128	12/96 – 09/02	32	4
				128	12/96 – 09/02	64	4
	20	49	6152	128	01/97 – now	128	12
				128	01/97 – 11/05	32	12
				80	06/05 – now	80	0
	21	5	544	128	10/98 – 12/04	128	4
				32	01/98 – 12/04	16	4
				128	11/02 – now	64	4
				128	11/05 – 12/04	32	4
H	22	1	256	256	11/04 – now	1024	0

Table 1. *Overview of systems. Systems 1–18 are SMP-based, and systems 19–22 are NUMA-based.*

What are common root causes of failures?



Relative frequency of root cause by system type.

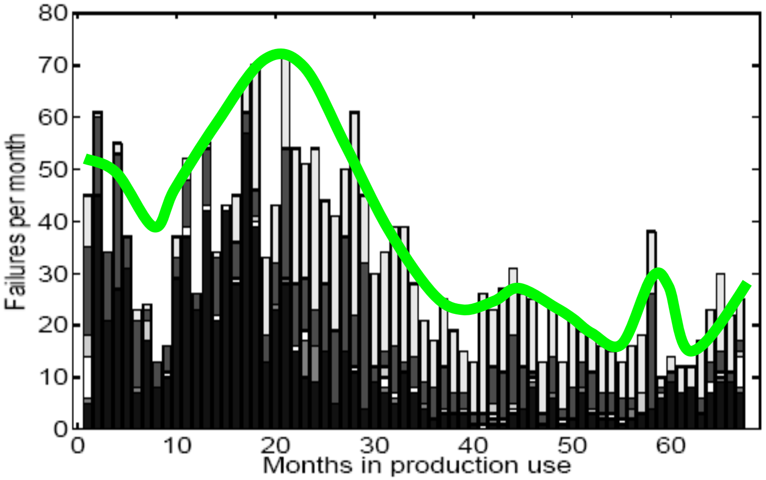


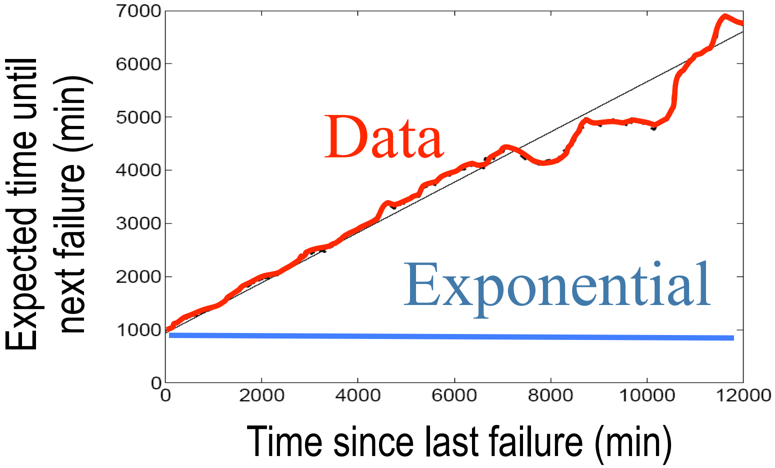
Fraction of total repair time caused by each root cause.

- Breakdown varies across systems
- Hardware and software most common root cause, and largest contributors to repair times

What do failure distributions look like?

- Failure rate with age does not always follow the traditional "bathtub"
  - Infant mortality may be seen for long into nominal lifetime
  - Steady state often not steady
- Time between failures in cluster not exponentially distributed
  - Much more variable
  - Time til next failure grows with time since last failure

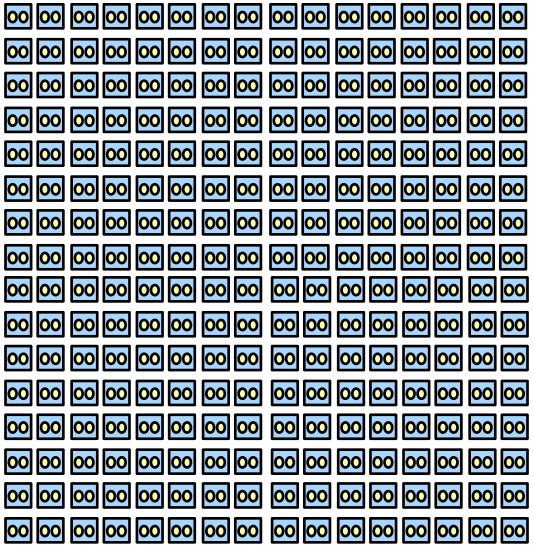




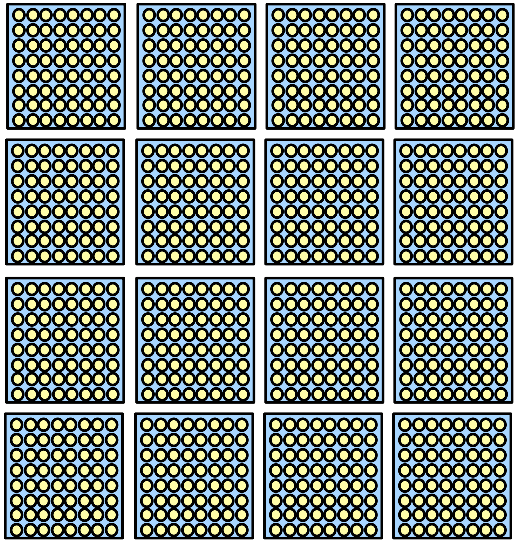
Carnegie Mellon
Parallel Data Laboratory

pdsi

LANL data has low & high density



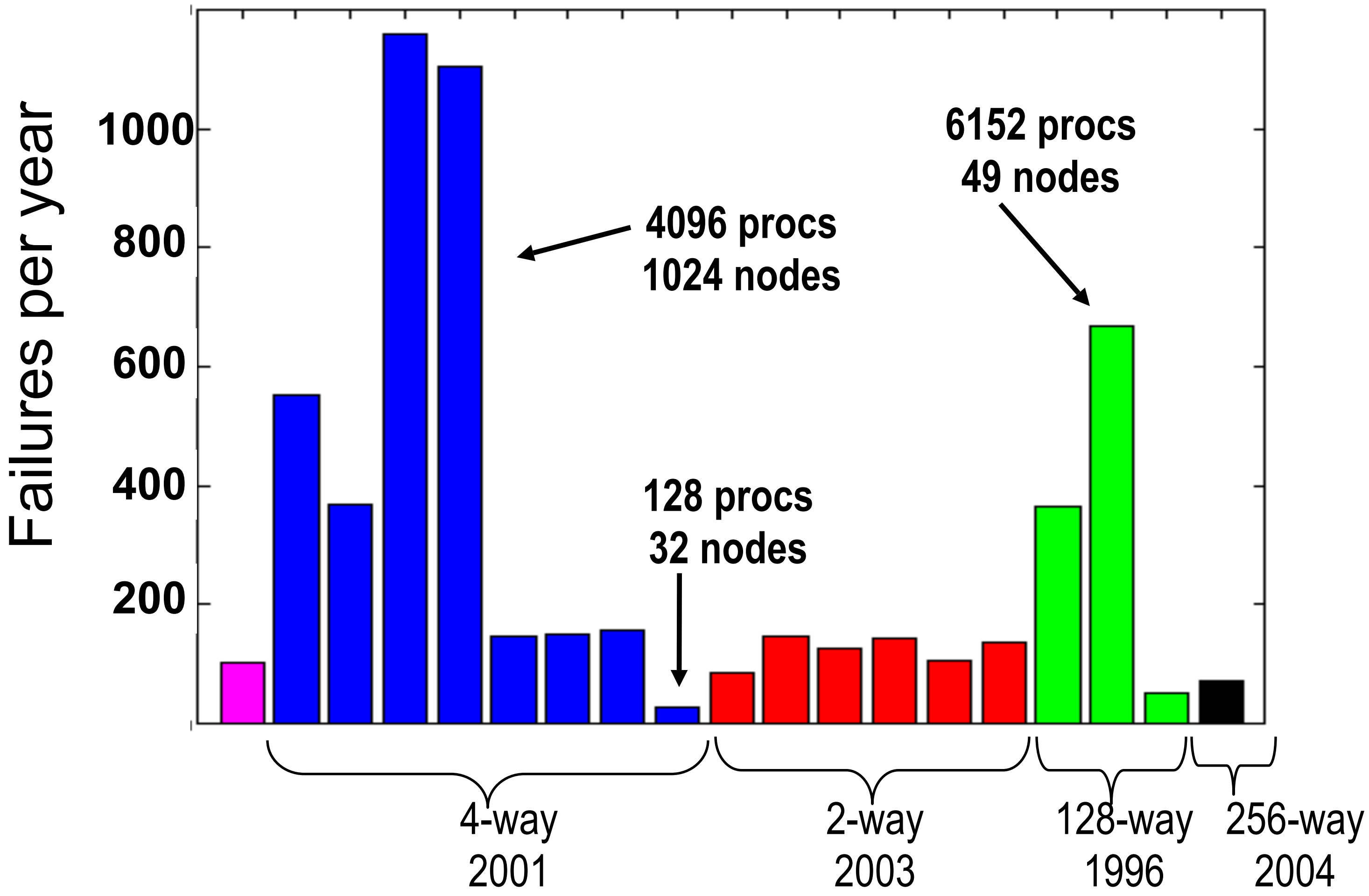
Clusters of 2/4-way **SMPs**
- **commodity components**
- 100s to 1000s of nodes.



Clusters of **NUMAs**
- **128-256 procs per node**
- 10s of nodes.

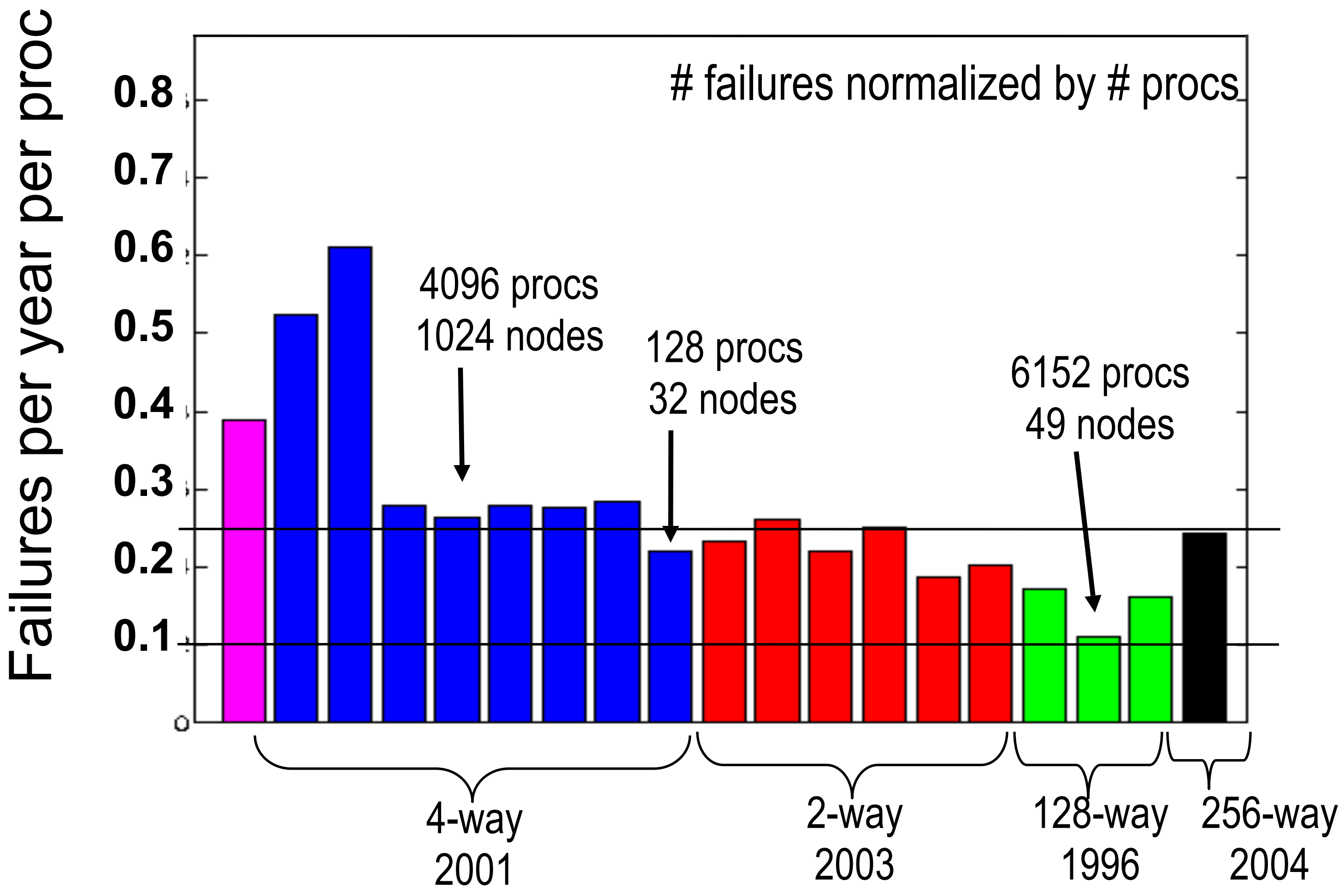
- Interruptions proportional to nodes? OSes? Procs?

System failure rate highly variable



Carnegie Mellon
Parallel Data Laboratory

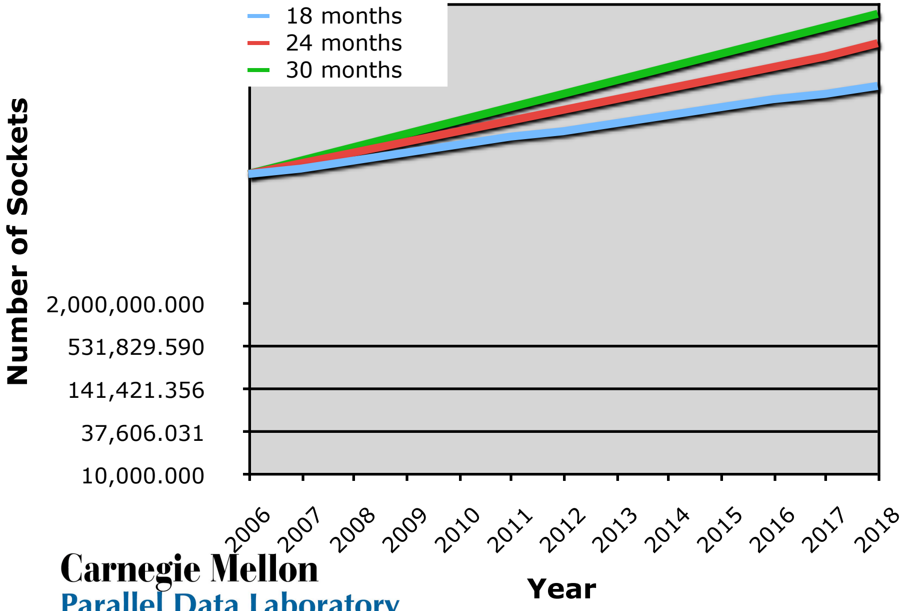
Best model: failures track # of processor chips

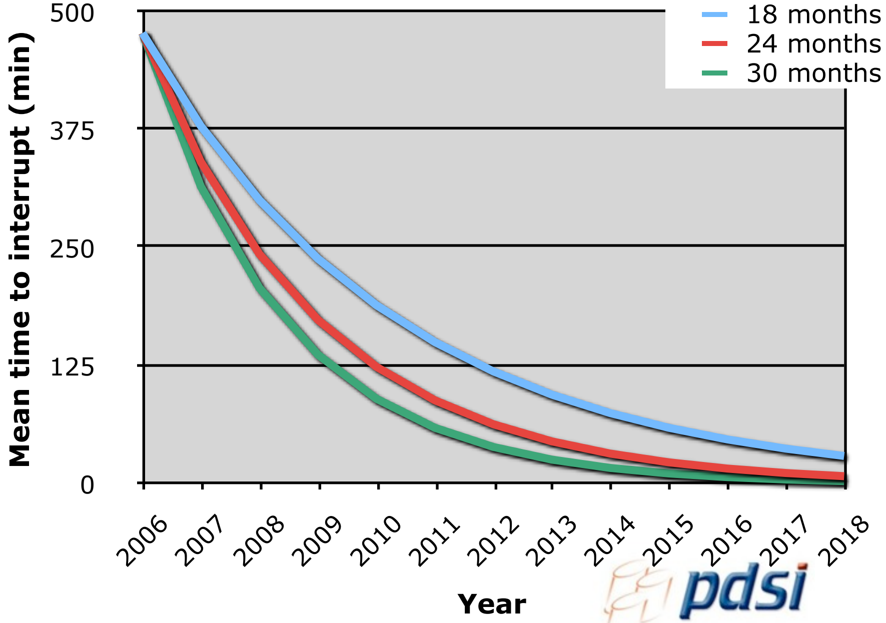


Carnegie Mellon
Parallel Data Laboratory

Petascale projections: more failures

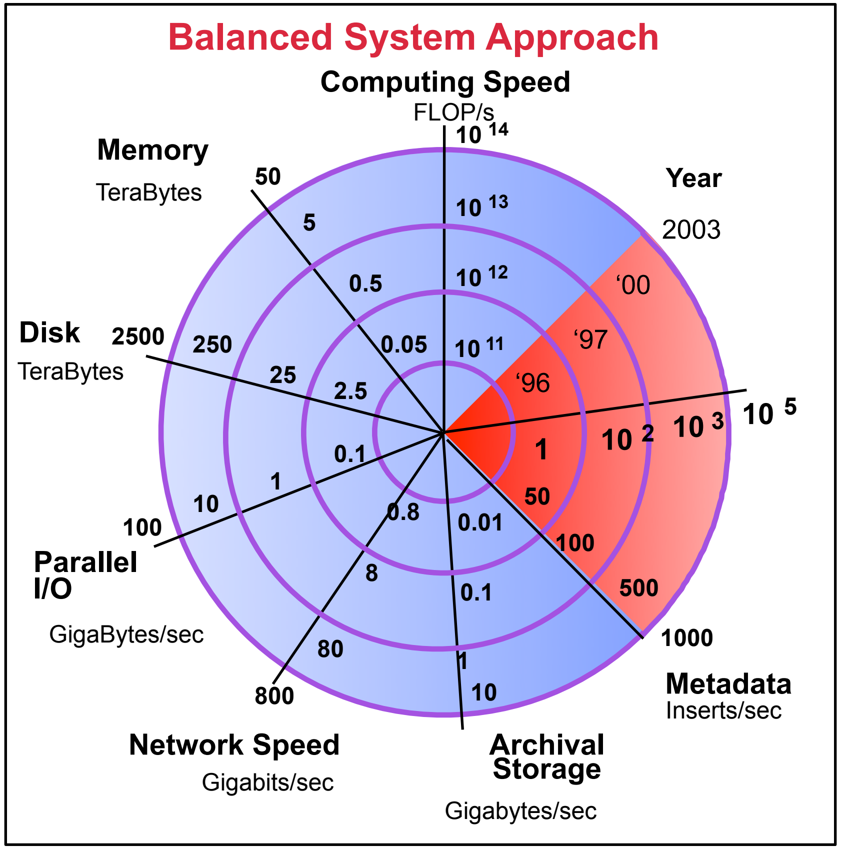
- Con't top500.org annual 2X peak FLOPS
  - Set to 1 PF plan for ORNL Baker, LANL Roadrunner in 2008
- Cycle time flat; Cores/chip on Moore's law
  - Consider 2X cores per chip every 18, 24, 30 months
- # sockets, 1/MTTI = failure rate up 25%-50% per year
  - Optimistic 0.1 failures per year per socket (vs. historic 0.25)





Checkpointing for app failure tolerance

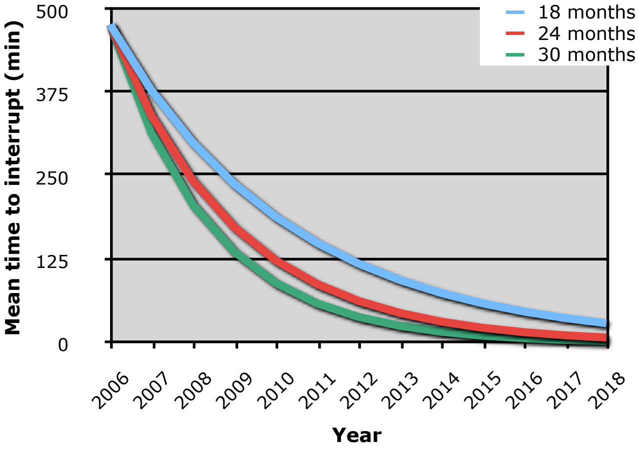
- Periodic (p) pause to capture checkpoint (t)
- On failure, roll back & restart from checkpoint
- Driven by tight coupling of parallel processes, esp. memory intensive
- Balanced systems
  - Memory size tracks FLOPS
  - Disk speed tracks both
  - Checkpoint capture (t) constant
  - 1 - App util = t / p + p / (2 * MTTI); $p^2 = 2 * t * MTTI$
  - If MTTI was constant, app utilization would be too

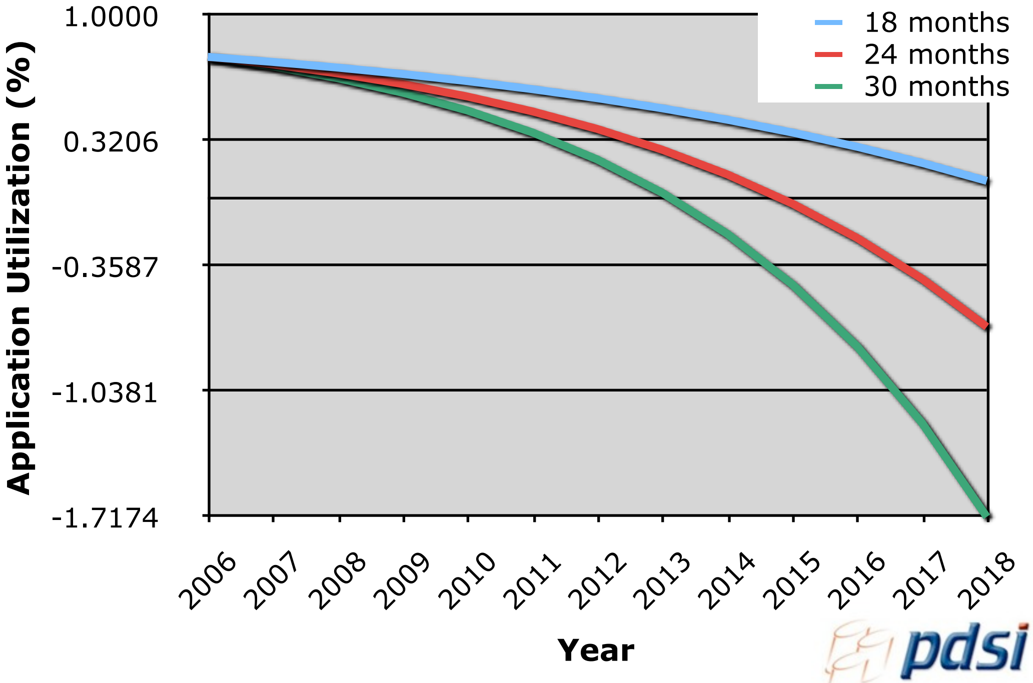


Carnegie Mellon
Parallel Data Laboratory

More failures hurts app's utilization

- Balanced: Mem, disk speed track FLOPS (constant t)
  - 1 - App util = t / p + p / (2 * MTTI); p^2 = 2 * t * MTTI
  - Since MTTI is dropping, checkpoint interval drops,
- So Application utilization drops progressively faster
- Half machine gone soon and exascale era bleak
- Not acceptable



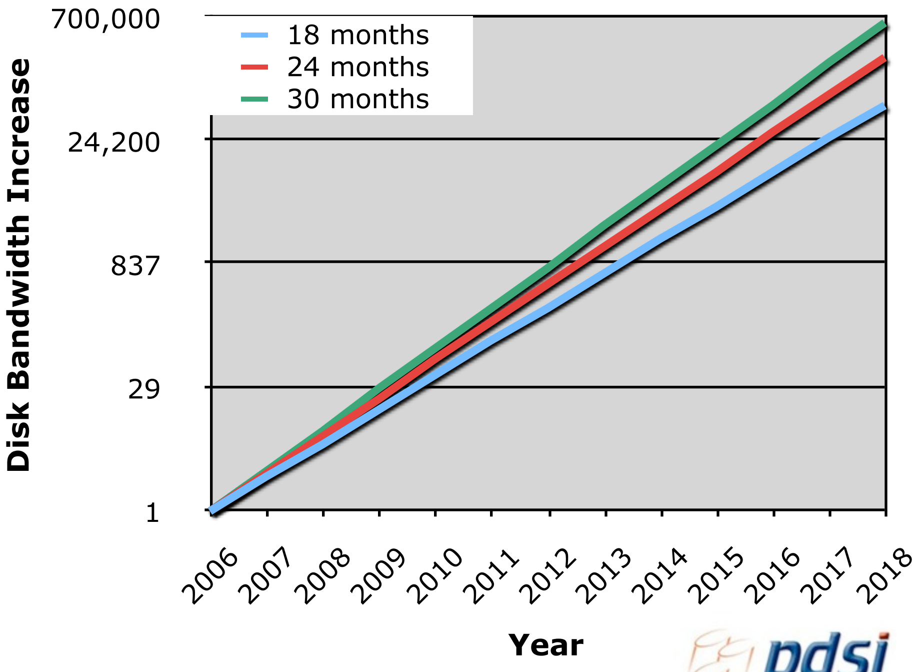


Carnegie Mellon
Parallel Data Laboratory

pdsi

Storage bandwidth to the rescue?

- Increase storage bandwidth to counter for MTTI?
- First, balance requires storage bandwidth track FLOPS, 2X per year, but disks 20% faster each year
  - Number of disks up 67% each year just for balance !
- To also counter MTTI trend
  - # 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)

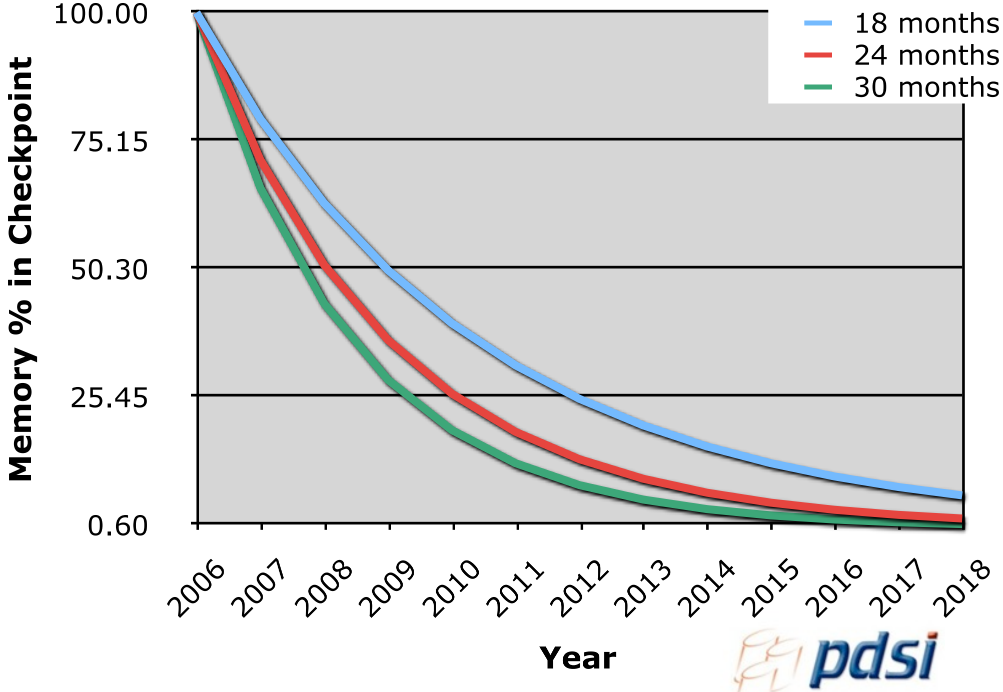


Smaller applications escape

- If an app uses 1/n of machine (sockets & memory)
  - 1 - App util = t/n / p + p / (2 * n*MTTI); p^2 = 2 * t/n * n*MTTI
  - Checkpoint overhead of subset resources is reduced by n
  - Assume full storage bandwidth avail for small checkpoint
- If app uses constant resources, it counters MTTI
  - ie., less and less of biggest machine
- Peak machines, when sliced up, see less inefficiency
- But Hero Apps, those that motivate ever bigger machines, gain nothing
  - Hero Apps are primary target of revisiting checkpoint/restart

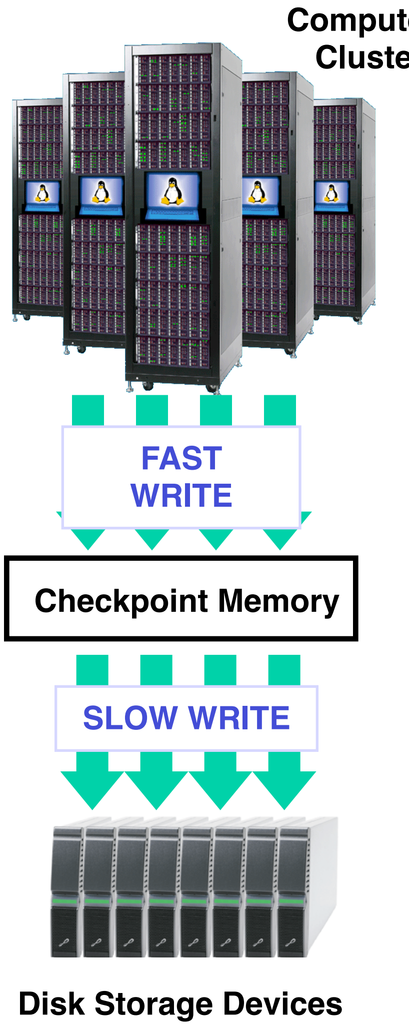
Applications squeeze checkpoints?

- So far, assumed checkpoint size is memory
- Could Apps counter MTTI with compression?
  - Lots of cycles for compression when saturating storage
- Size of checkpoint has to decrease with MTTI
  - Smaller fraction of memory with each machine
  - Drop 25-50% per year
- If possible ….
- Cache checkpoint in other node's memory
- Decrease pressure on storage bandwidth and storage costs



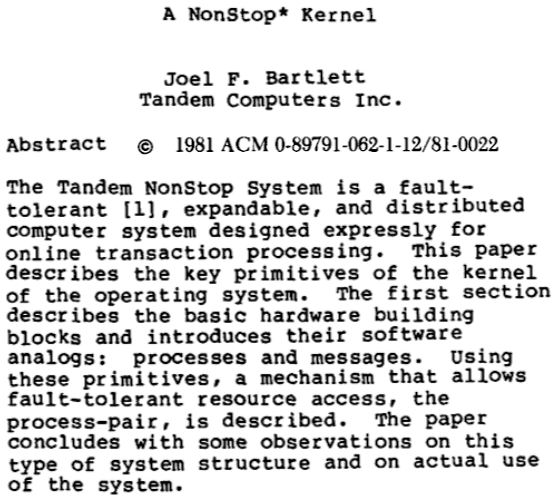
Dedicated memory devices?

- Use memory to stage checkpoint
  - Fast write from node to stage memory
    - Short checkpoint capture time
  - Slower write from stage to disk
    - Finish before next checkpoint
- Where is checkpoint memory
  - Different fault domain from node memory
  - Can wrap onto other nodes, but “slow” writing is constant OS noise for compute
  - Limited by networking; will be parallel
  - Probably CPU-light compute nodes
- Maybe more costly than storage solution
  - Starts by doubling, or more, memory size
  - Maybe Flash if used only for checkpoints



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 of pair synch
- No stopping to checkpoint
  - Less pressure on storage bandwidth except for visualization checkpoints

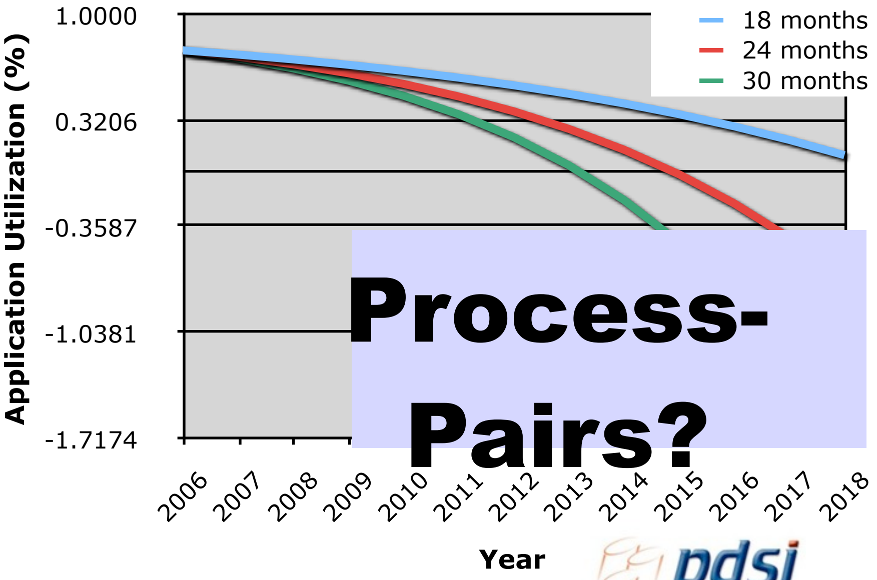


A NonStop* Kernel

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.



Process-Pairs?

Recap so far

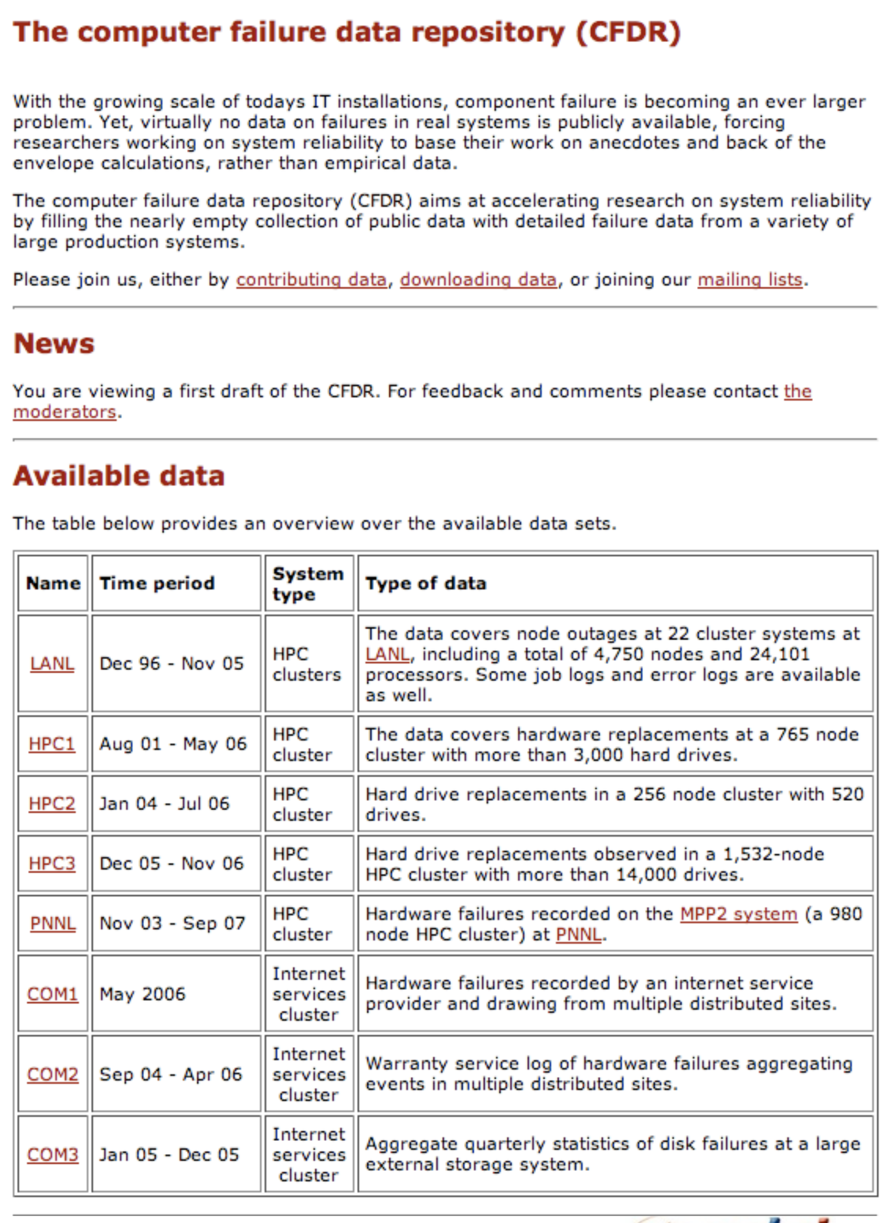
- Failure rates proportional to number of components
  - Specifically, growing # sockets in parallel computer
- If peak compute continues to outstrip Moore's law
  - MTTI will drop, forcing more checkpoints & restarts
- Hero apps, wanting all the resources, bear burden
  - Storage won't keep up b/c cost; dedicated device similar
  - Squeezing checkpoint not believable; process pairs is

- Schroeder, B., G. A. Gibson, "Understanding Failures in Petascale Computers," *Journal of Physics: Conference Series* **78** (2007), SciDAC 2007.

garth@cs.cmu.edu & www.pdsi-scidac.org

CFDR

- Gather & publish real failure data of computing at scale
- Community effort
  - USENIX clearinghouse
  - http://cfdr.usenix.org/
- Storage, networks, computers, etc
- Anonymized as needed
- Educate researchers
- DSN06, FAST07 papers
  - www.pdl.cmu.edu/FailureData/



The computer failure data repository (CFDR)

With the growing scale of todays 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.

Please join us, either by contributing data, downloading data, or joining our mailing lists.

News

You are viewing a first draft of the CFDR. For feedback and comments please contact the moderators.

Available data

The table below provides an overview over the available data sets.

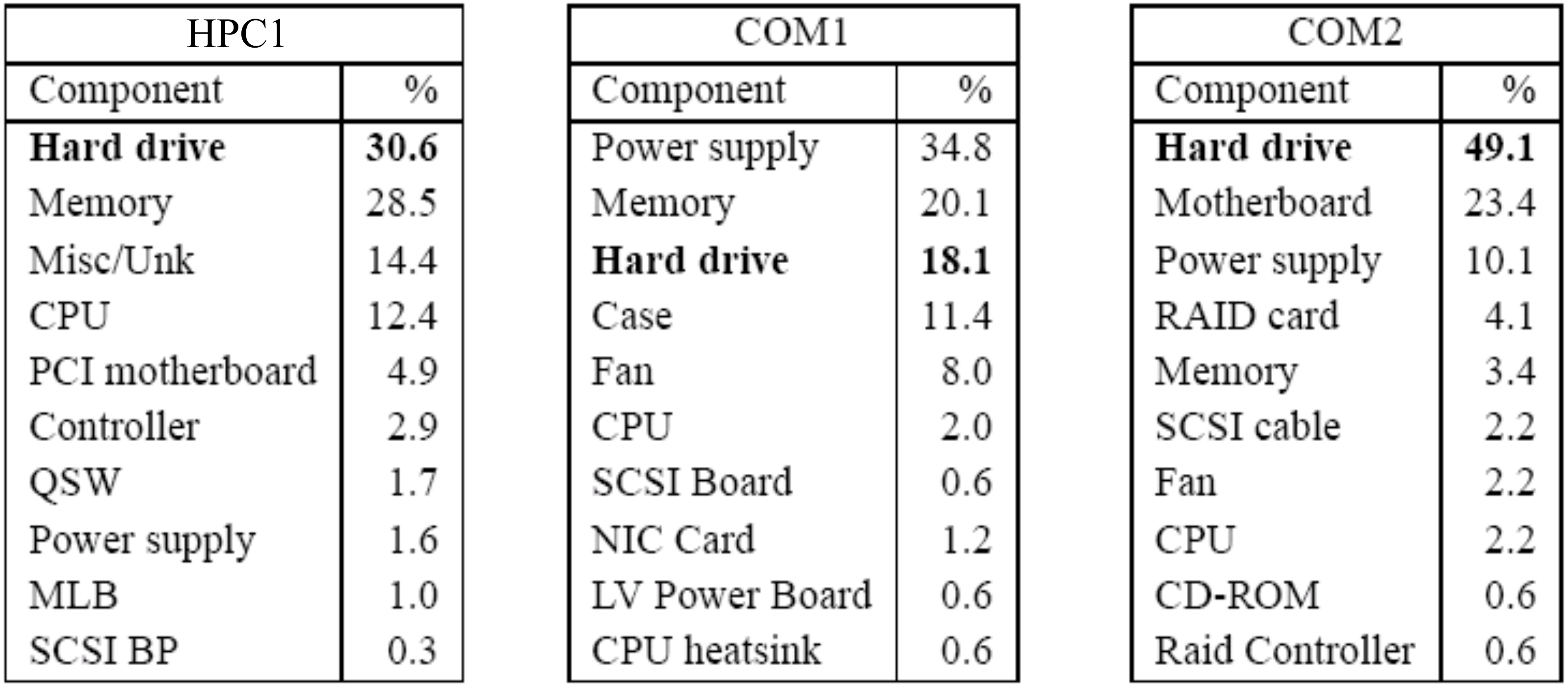
Name	Time period	System type	Type of data
LANL	Dec 96 - Nov 05	HPC clusters	The data covers node outages at 22 cluster systems at LANL, including a total of 4,750 nodes and 24,101 processors. Some job logs and error logs are available as well.
HPC1	Aug 01 - May 06	HPC cluster	The data covers hardware 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.
PNNL	Nov 03 - Sep 07	HPC cluster	Hardware failures recorded on the MPP2 system (a 980 node HPC cluster) at PNNL.
COM1	May 2006	Internet services cluster	Hardware failures recorded by an internet service provider and drawing from multiple distributed sites.
COM2	Sep 04 - Apr 06	Internet services cluster	Warranty service log of hardware failures aggregating events in multiple distributed sites.
COM3	Jan 05 - Dec 05	Internet services cluster	Aggregate quarterly statistics of disk failures at a large external storage system.

Failure data: hardware replacement logs

		Type of drive	Count	Duration
Pittsburgh Supercomputing Center	HPC1	18GB 10K RPM SCSI 36GB 10K RPM SCSI	3,400	5 yrs
Los Alamos National Laboratory EST. 1943	HPC2	36GB 10K RPM SCSI	520	2.5 yrs
Supercomputing X	HPC3	15K RPM SCSI 15K RPM SCSI 7.2K RPM SATA	14,208	1 yr
Various HPCs	HPC4	250GB SATA 500GB SATA 400GB SATA	13,634	3 yrs
Internet services Y	COM1	10K RPM SCSI	26,734	1 month
	COM2	15K RPM SCSI	39,039	1.5 yrs
	COM3	10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL	3,700	1 yr

Relative frequency of disk replacements

The top ten of replaced components



HPC1	
Component	%
Hard drive	**30.6**
Memory	28.5
Misc/Unk	14.4
CPU	12.4
PCI motherboard	4.9
Controller	2.9
QSW	1.7
Power supply	1.6
MLB	1.0
SCSI BP	0.3

COM1	
Component	%
Power supply	34.8
Memory	20.1
Hard drive	**18.1**
Case	11.4
Fan	8.0
CPU	2.0
SCSI Board	0.6
NIC Card	1.2
LV Power Board	0.6
CPU heatsink	0.6

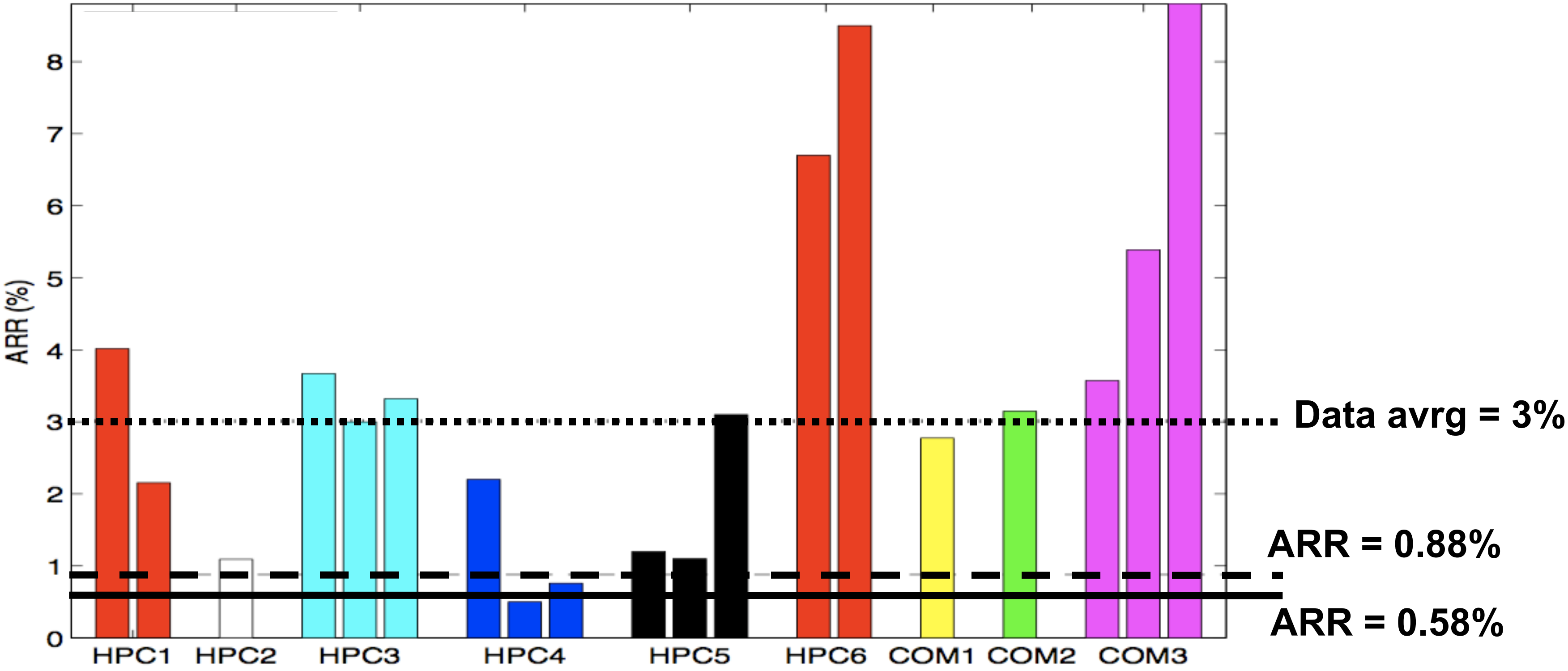
COM2	
Component	%
Hard drive	**49.1**
Motherboard	23.4
Power supply	10.1
RAID card	4.1
Memory	3.4
SCSI cable	2.2
Fan	2.2
CPU	2.2
CD-ROM	0.6
Raid Controller	0.6

- All hardware fails, though disks failures often common

Annual disk replacement rate (ARR)

- Datasheet MTTFs are 1,000,000 to 1,500,000 hours.

=> Expected annual replacement rate (ARR): 0.58 - 0.88 %.

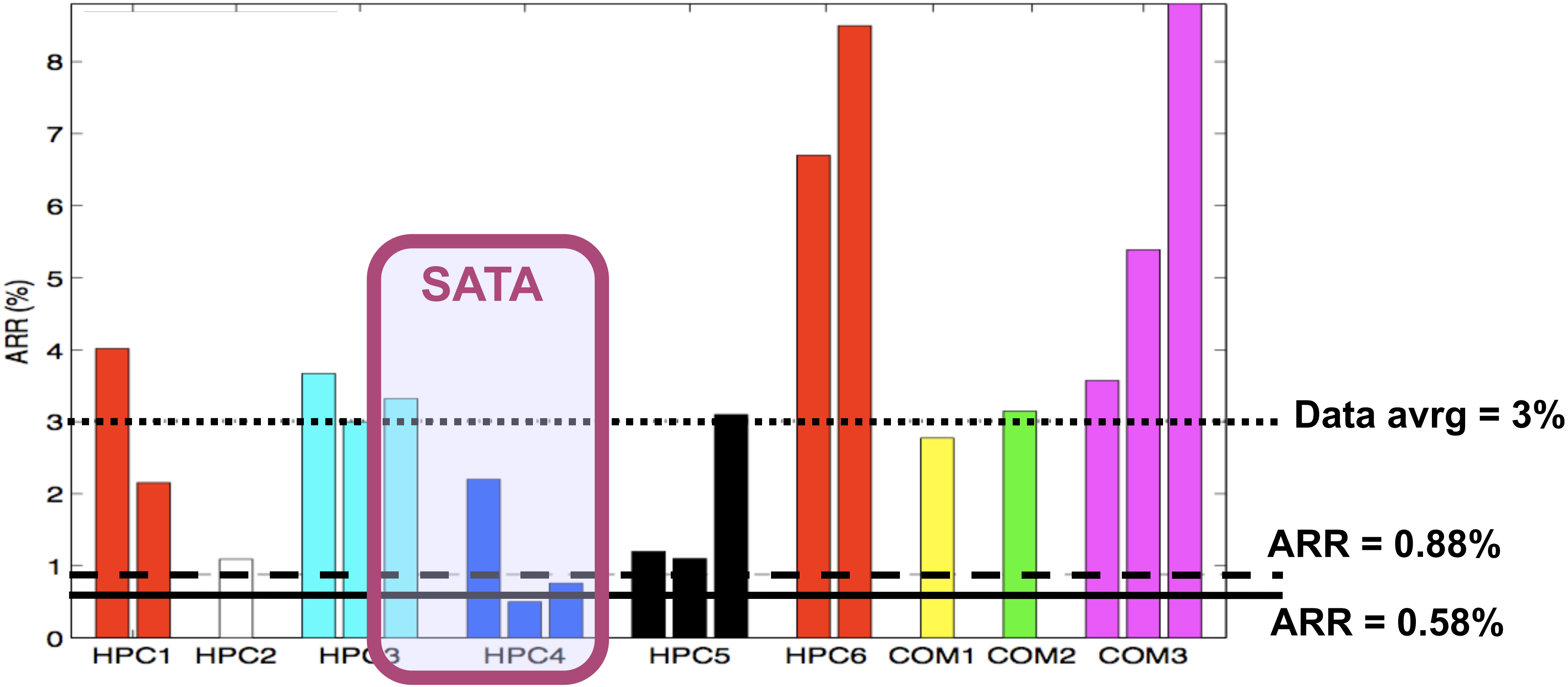


Carnegie Mellon
Parallel Data Laboratory

Annual disk replacement rate (ARR)

- Datasheet MTTFs are 1,000,000 to 1,500,000 hours.

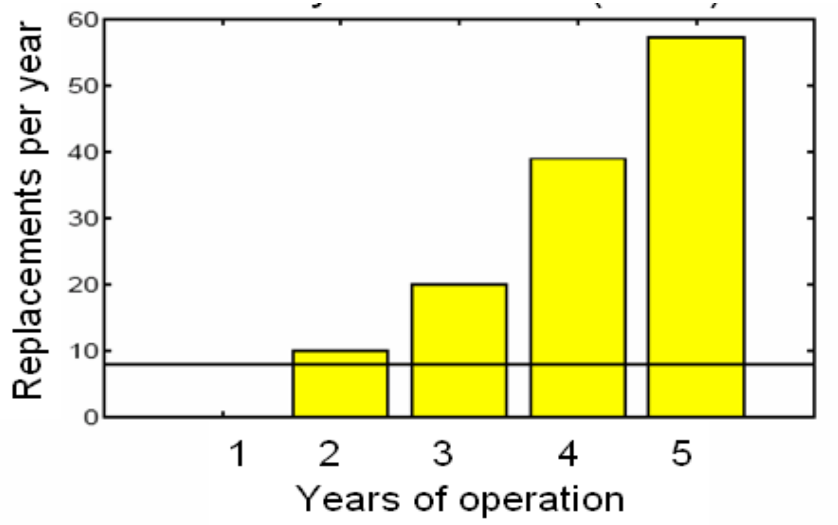
=> Expected annual replacement rate (ARR): 0.58 - 0.88 %.



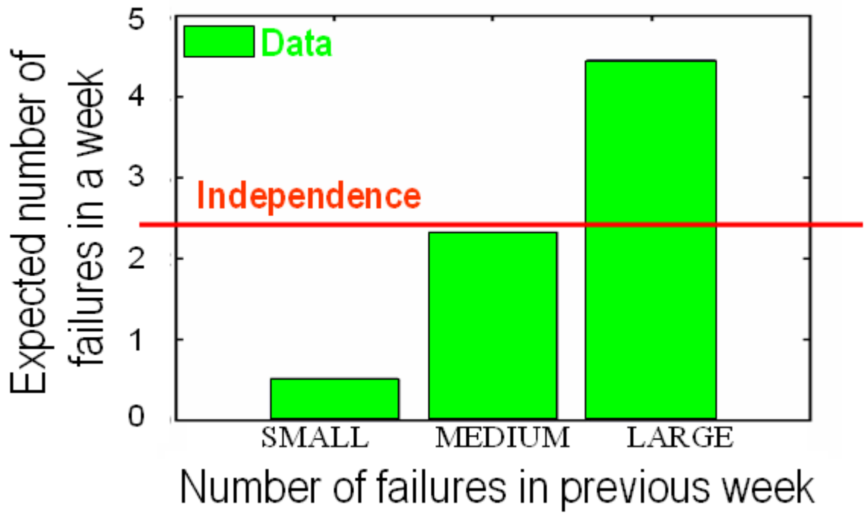
- Poor evidence for SATA fail rates higher than SCSI or FC

What do failure distributions look like?

- Failure rate with age does not follow the traditional "bathtub"
  - Infant mortality is mostly not seen by customers
  - Wear out often prominent effect

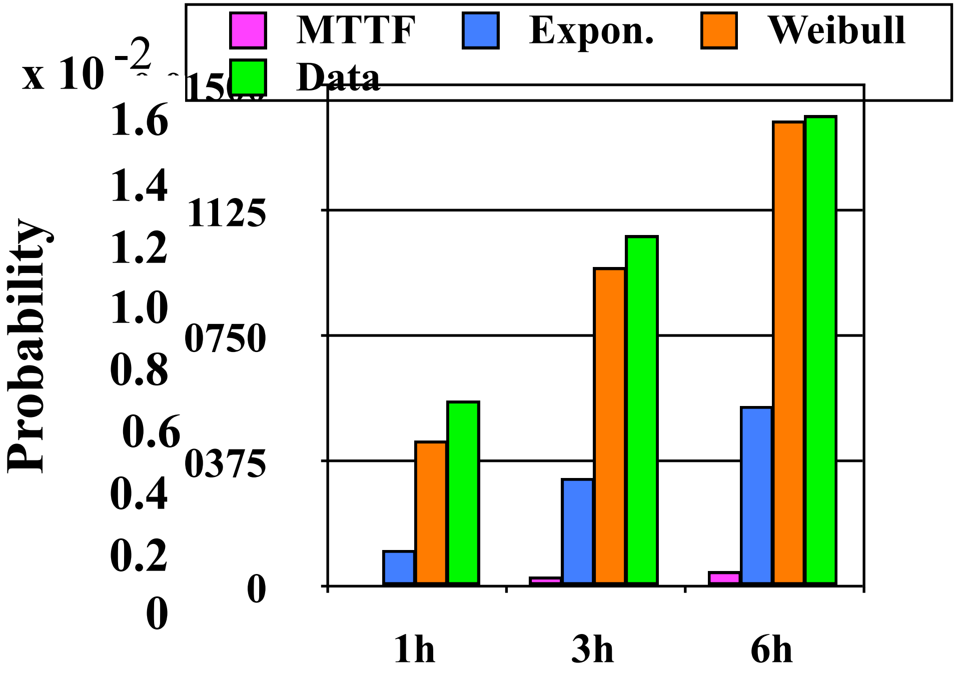


- Failures significantly clustered
  - Weeks of few/many failures predict few/many failures next week



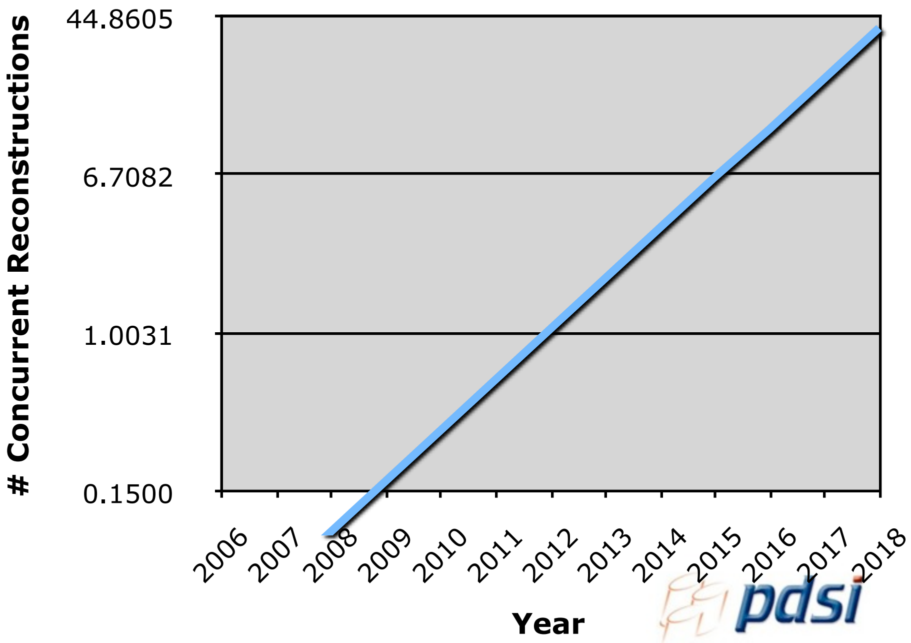
Non-exponential disk failures

- RAID failure depends on probability of a 2nd disk failure
  - during reconstruction (typically 10, growing to 100 hours)
- What is probability of a 2nd disk failure in the real world?
  - Need more than field failure rates, need measure of burstiness



While on storage issues …

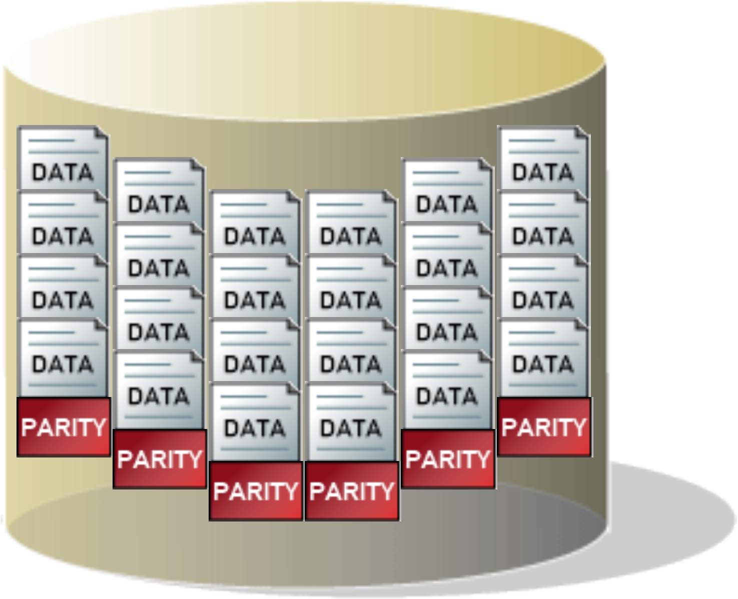
- Balanced disk bandwidth: more disks & disk failures
- RAID (level 5, 6 or stronger codes) protect data
  - At cost of online reconstruction of all lost data
  - Larger disks: longer reconstructions, hours become days
- Consider # concurrent reconstructions
- 10-20% now, but ….
- Soon 100s of concurrent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



And point solutions for narrow problems

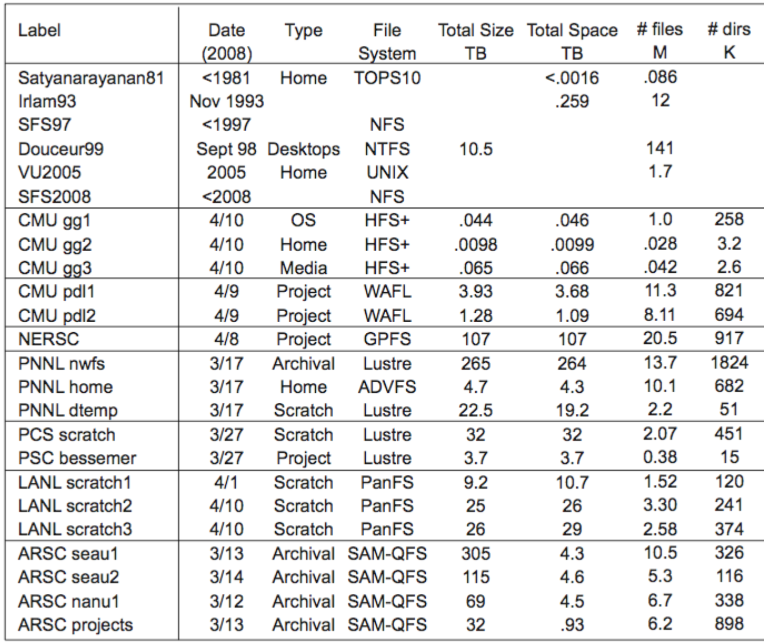
- Study media errors
- Devise per disk correcting codes to scale with disk size
  - Improves on internal ECC capabilities (limited by economics
- Independent of traditional cross disk parity schemes
- Avoids using double failed disk codes until double failures are the problem

Example: Panasas Vertical Parity

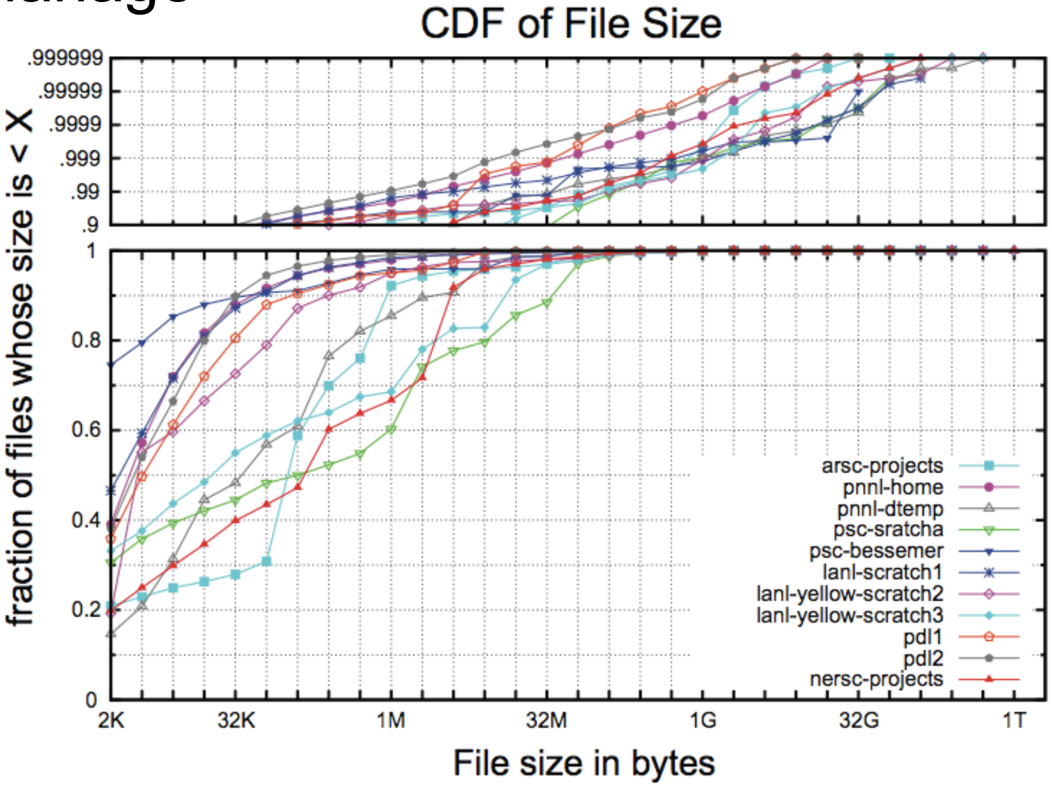


And it ain’t all huge data files

- Study data distributions – millions of files – 20-80% tiny
  - Still majority of space in relatively few huge files (like checkpoints)
- Lots more metadata to manage



Label	Date (2008)	Type	File System	Total Size TB	Total Space TB	# files M	# dirs K
Satyanarayanan81	<1981	Home	TOPS10		<.0016	.086	
Irlam93	Nov 1993				.259	12	
SFS97	<1997		NFS				
Douceur99	Sept 98	Desktops	NTFS	10.5		141	
VU2005	2005	Home	UNIX			1.7	
SFS2008	<2008		NFS				
CMU gg1	4/10	OS	HFS+	.044	.046	1.0	258
CMU gg2	4/10	Home	HFS+	.0098	.0099	.028	3.2
CMU gg3	4/10	Media	HFS+	.065	.066	.042	2.6
CMU pdl1	4/9	Project	WAFL	3.93	3.68	11.3	821
CMU pdl2	4/9	Project	WAFL	1.28	1.09	8.11	694
NERSC	4/8	Project	GPFS	107	107	20.5	917
PNNL nwfs	3/17	Archival	Lustre	265	264	13.7	1824
PNNL home	3/17	Home	ADVFS	4.7	4.3	10.1	682
PNNL dtemp	3/17	Scratch	Lustre	22.5	19.2	2.2	51
PCS scratch	3/27	Scratch	Lustre	32	32	2.07	451
PSC bessemer	3/27	Project	Lustre	3.7	3.7	0.38	15
LANL scratch1	4/1	Scratch	PanFS	9.2	10.7	1.52	120
LANL scratch2	4/10	Scratch	PanFS	25	26	3.30	241
LANL scratch3	4/10	Scratch	PanFS	26	29	2.58	374
ARSC seau1	3/13	Archival	SAM-QFS	305	4.3	10.5	326
ARSC seau2	3/14	Archival	SAM-QFS	115	4.6	5.3	116
ARSC nanu1	3/12	Archival	SAM-QFS	69	4.5	6.7	338
ARSC projects	3/13	Archival	SAM-QFS	32	.93	6.2	898



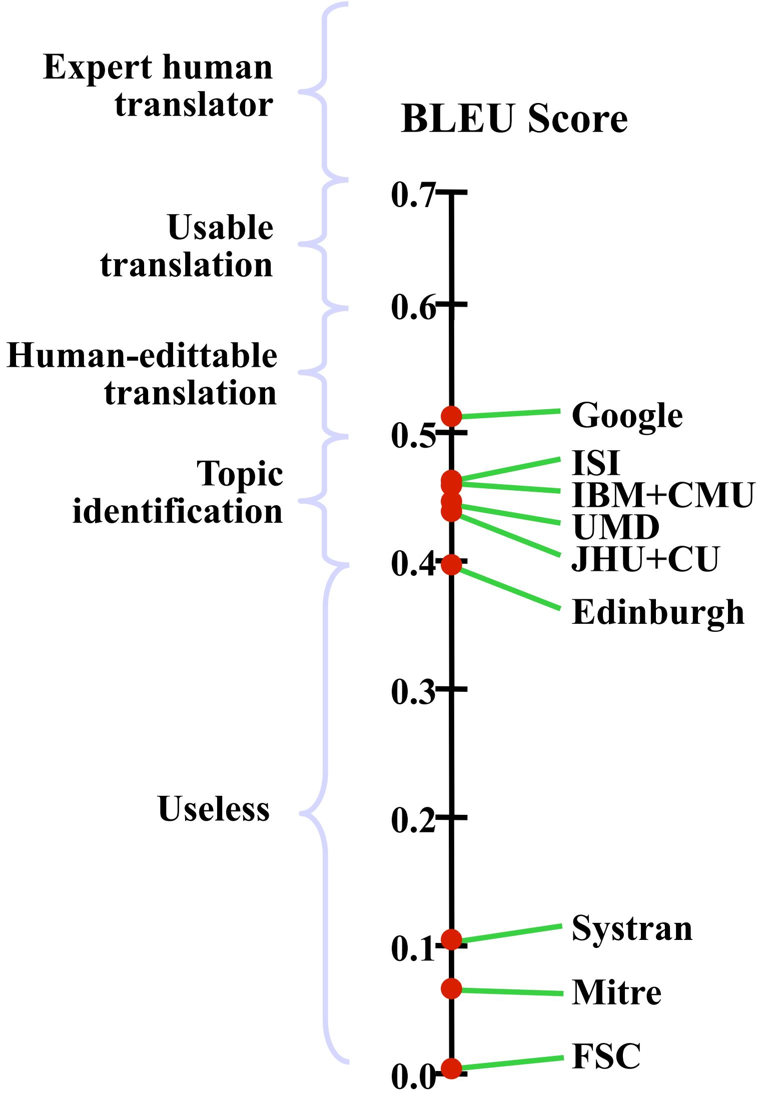
CDF of File Size

Closing

- Future parallel computing increasingly suffers failures
- Field data needs to be collected and shared
  - cfdr.usenix.org: please use and contribute
- Traditional fault tolerance needs to be revisited
  - Checkpointing needs new paradigms
- Systems need to be designed to operate in repair
  - Storage may be always repairing multiple failed disks

www.pdsi-scidac.org

Data intensive computing has many forms



NIST translate 100 articles

- Arabic-English competition

2005 outcome: Google wins!

Qualitatively better on 1st entry

Not most sophisticated approach

Brute force statistics

But more data & compute !!

200M words from UN translations

1 trillion words of English grammar

1000 processor cluster

Science of all types going to scale

Can't do the best science without it