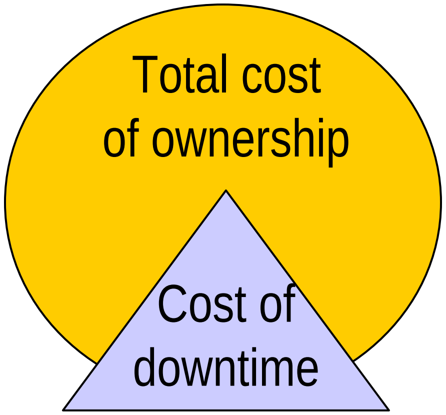
Failures in large systems: Collecting, analyzing, modeling and exploiting real data on failures in large systems.

Bianca Schroeder, Garth Gibson

Department of Computer Science
Carnegie Mellon University

Reliability is important

- Failures are expensive.



Total cost of ownership

Cost of downtime

- System flakiness is major source of user frustration:
  - 25% in survey have seen peers kicking their computers.

  - 2% claim to have hit the person next to them in their frustration.

Reliability and HEC

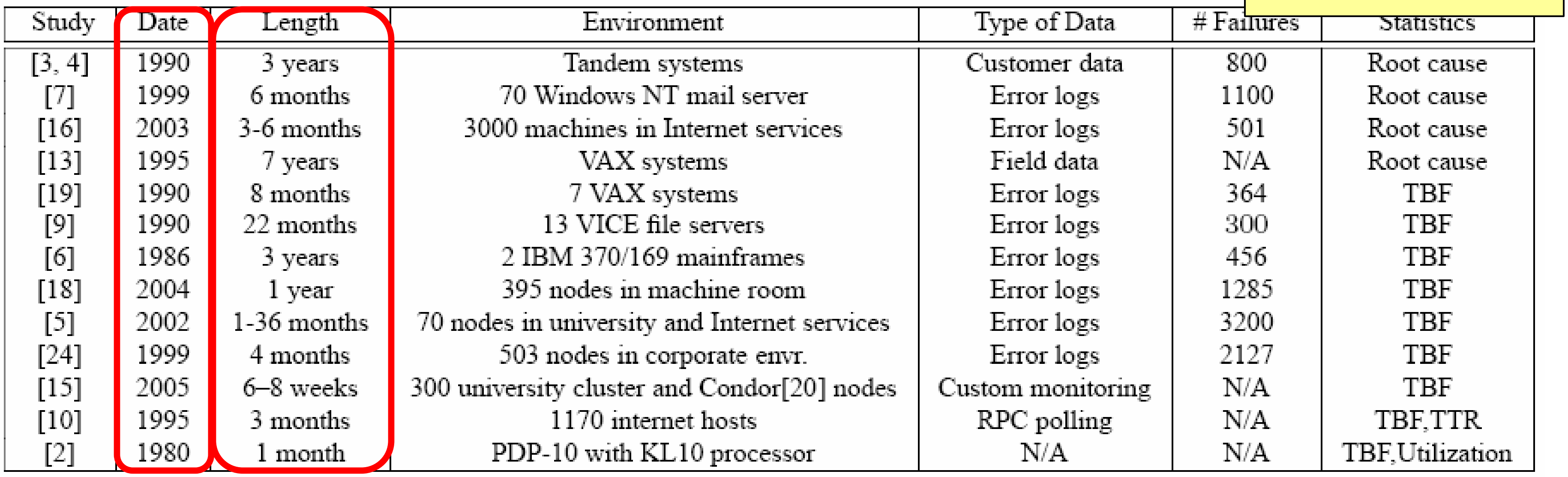
- ***Petascale computing*** is coming.
  - Orders of magnitude more components.
  - Orders of magnitude more ***failures*** **….**

What do failures look like?

- Making systems more reliable requires good understanding of real failures:
  - Cause of failures?
  - Failure rates?
  - Time to repair?
  - What parameters affect the above?

What do failures look like?

Previous work:



Study	Date	Length	Environment	Type of Data	# Failures	Statistics
[3, 4]	1990	3 years	Tandem systems	Customer data	800	Root cause
[7]	1999	6 months	70 Windows NT mail server	Error logs	1100	Root cause
[16]	2003	3-6 months	3000 machines in Internet services	Error logs	501	Root cause
[13]	1995	7 years	VAX systems	Field data	N/A	Root cause
[19]	1990	8 months	7 VAX systems	Error logs	364	TBF
[9]	1990	22 months	13 VICE file servers	Error logs	300	TBF
[6]	1986	3 years	2 IBM 370/169 mainframes	Error logs	456	TBF
[18]	2004	1 year	395 nodes in machine room	Error logs	1285	TBF
[5]	2002	1-36 months	70 nodes in university and Internet services	Error logs	3200	TBF
[24]	1999	4 months	503 nodes in corporate envr.	Error logs	2127	TBF
[15]	2005	6–8 weeks	300 university cluster and Condor[20] nodes	Custom monitoring	N/A	TBF
[10]	1995	3 months	1170 internet hosts	RPC polling	N/A	TBF,TTR
[2]	1980	1 month	PDP-10 with KL10 processor	N/A	N/A	TBF,Utilization

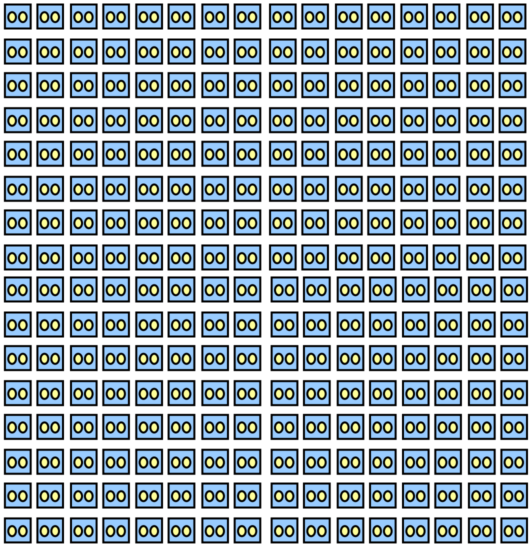
•None of the data publicly available!

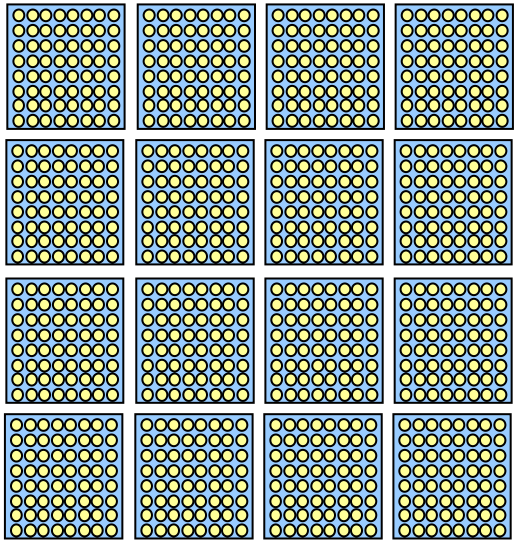
Talk outline

Publicly available!

- Our current work: Analysis of **9 years of failure data from LANL.**
- Long-term goals: Create public failure data repository.
  Exploit failure data for better system eval & design.

Typical LANL systems and workloads





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

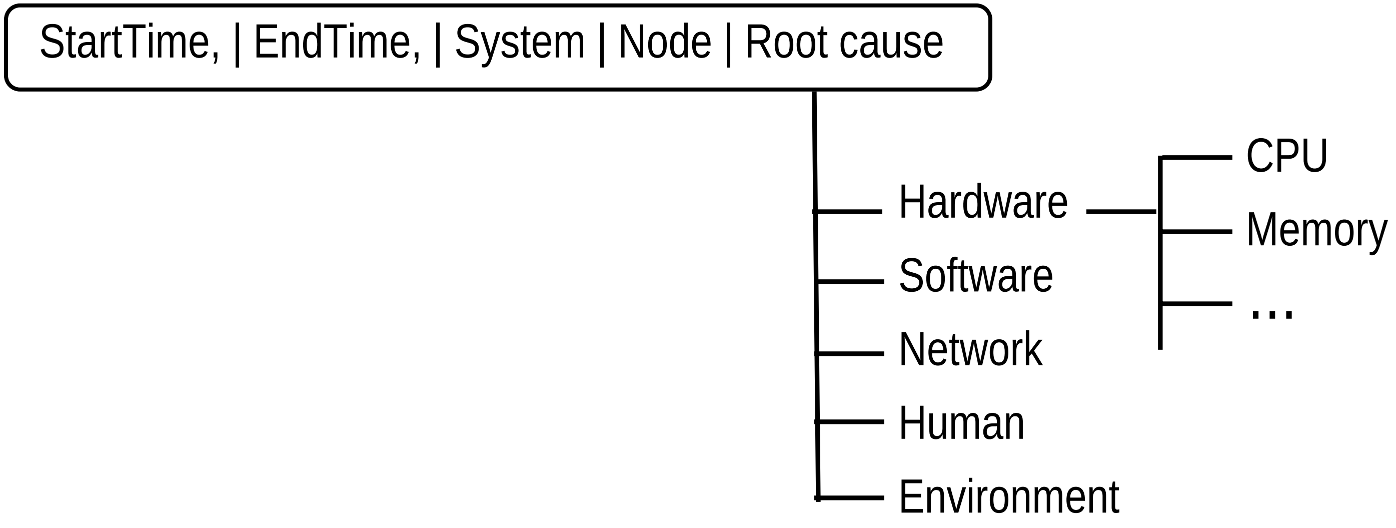
Clusters of NUMAs
- 128-256 procs per node
- 10s of nodes.

Workloads:

- Large-scale simulations and visualization, e.g. nuclear stockpile stewardship. Mostly CPU-bound.
- Failure tolerance through checkpoint-restart.

The data

- Record created by administrator for each node outage:



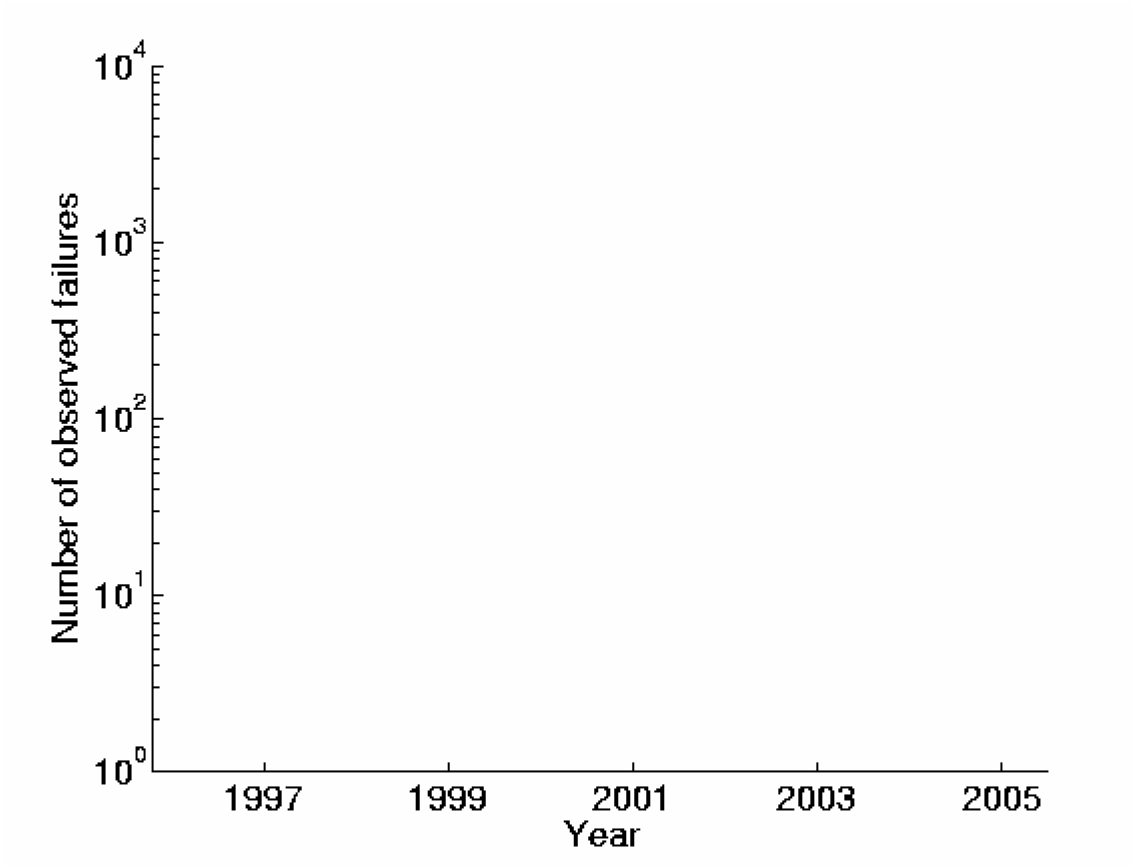
StartTime, | EndTime, | System | Node | Root cause

- Hardware
  - CPU
  - Memory
  - …
- Software
- Network
- Human
- Environment

The data

- Record created by administrator for each node outage:

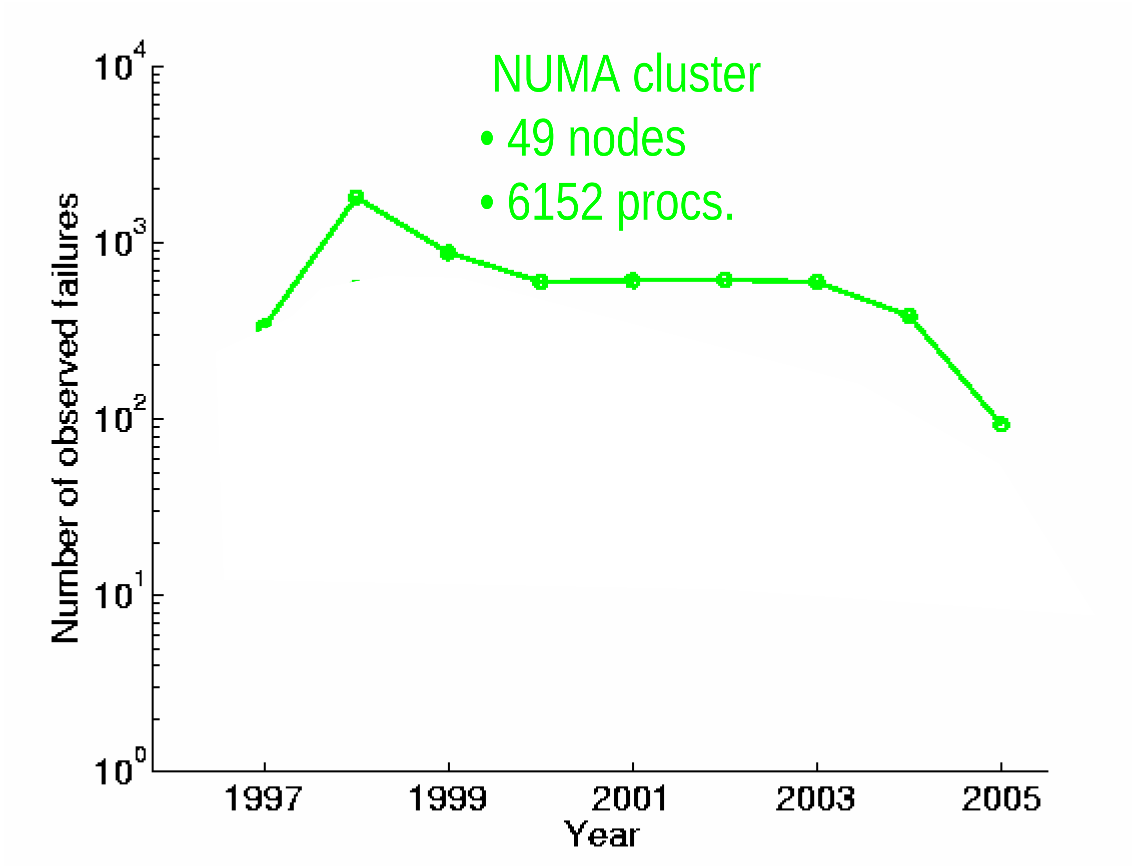
StartTime, | EndTime, | System | Node | Root cause



The data

- Record created by administrator for each node outage:

StartTime, | EndTime, | System | Node | Root cause

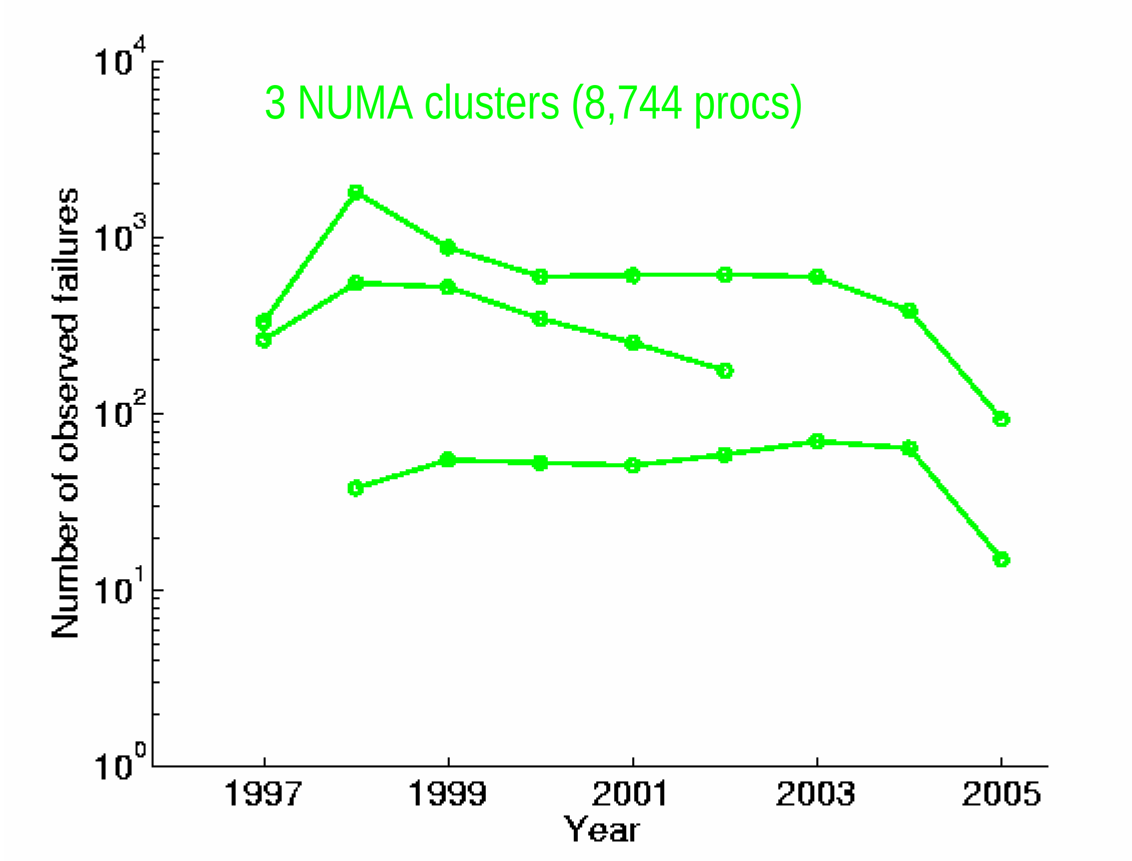


Carnegie Mellon
Parallel Data Laboratory

The data

- Record created by administrator for each node outage:

StartTime, | EndTime, | System | Node | Root cause

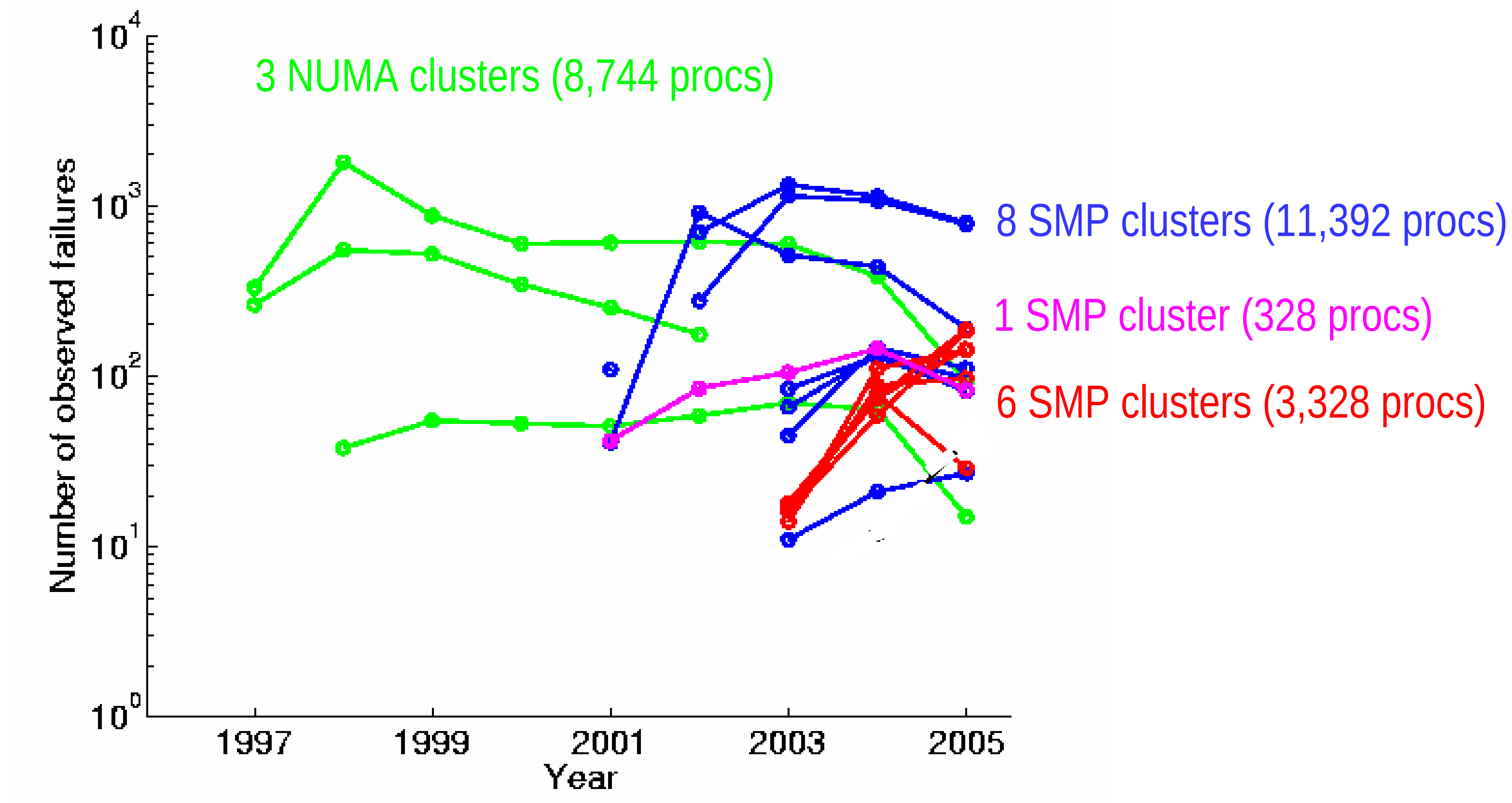


Carnegie Mellon
Parallel Data Laboratory

The data

- Record created by administrator for each node outage:

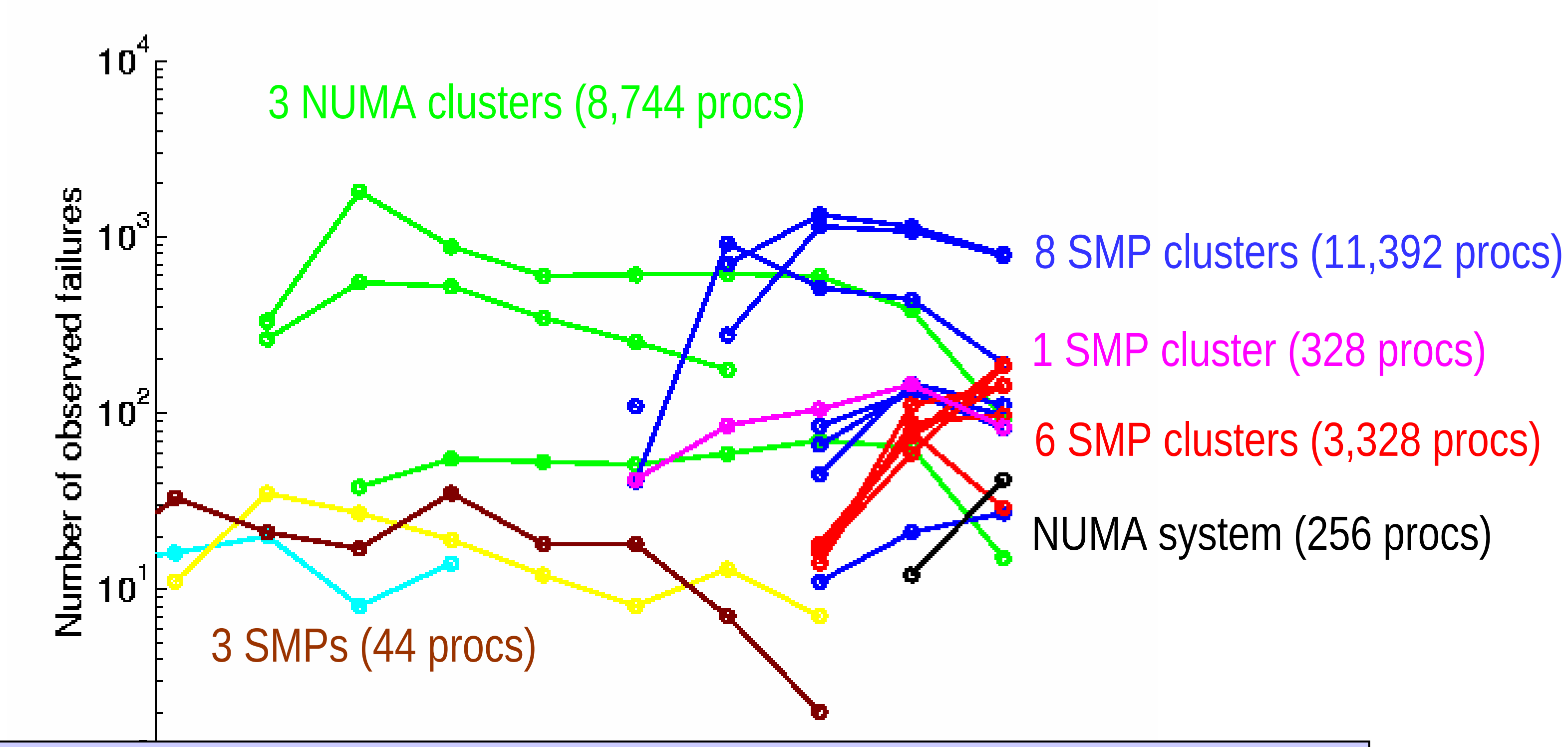
StartTime, | EndTime, | System | Node | Root cause



The data

- Record created by administrator for each node outage:

StartTime, | EndTime, | System | Node | Root cause

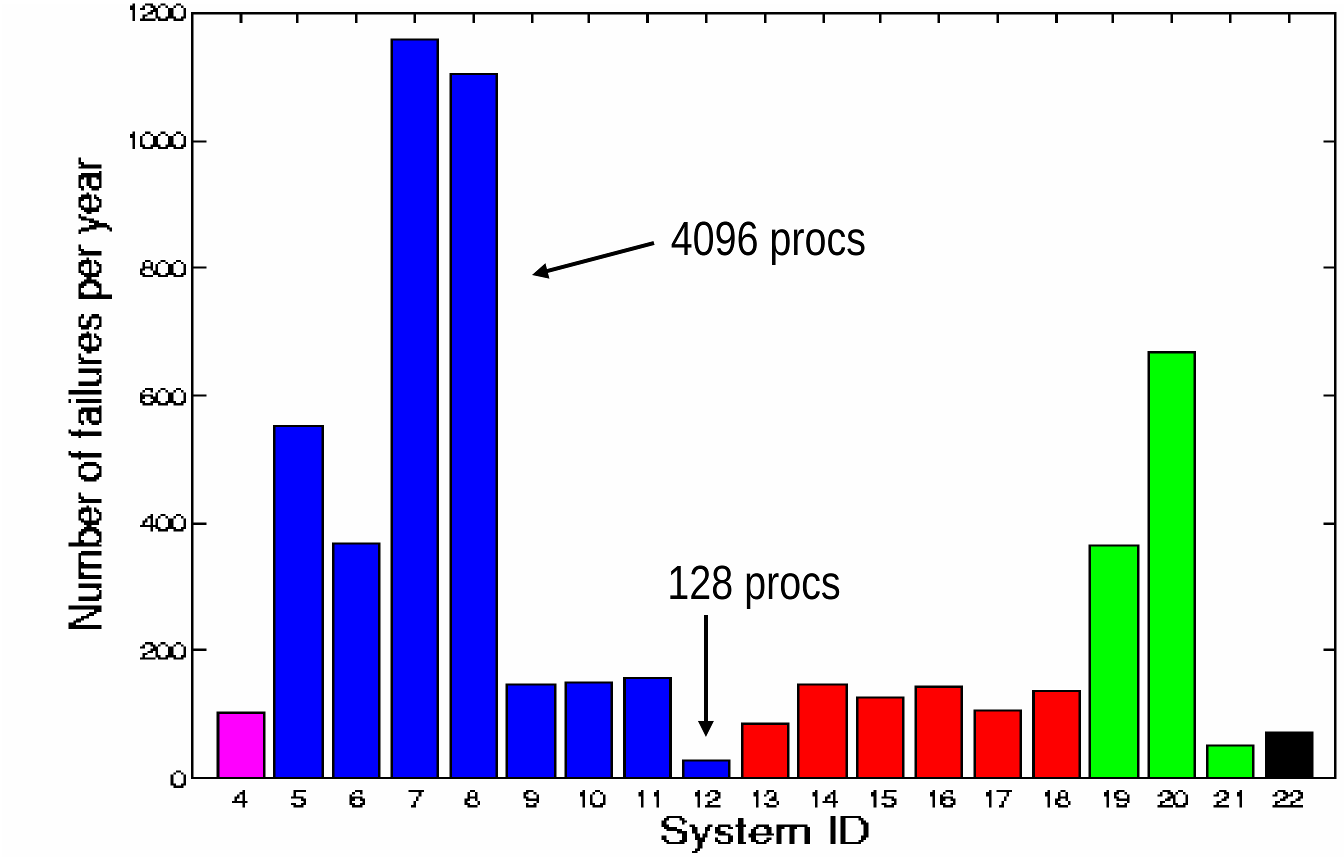


- 22 systems, 4,750 nodes and 24,101 processors.
- Total of **23,000 records** over **9 years**!

Outline

- What do **failure rates** (or time between failures) look like?
- What do **repair times** look like?
- What are the common **root causes** of failures?

What do failure rates look like?

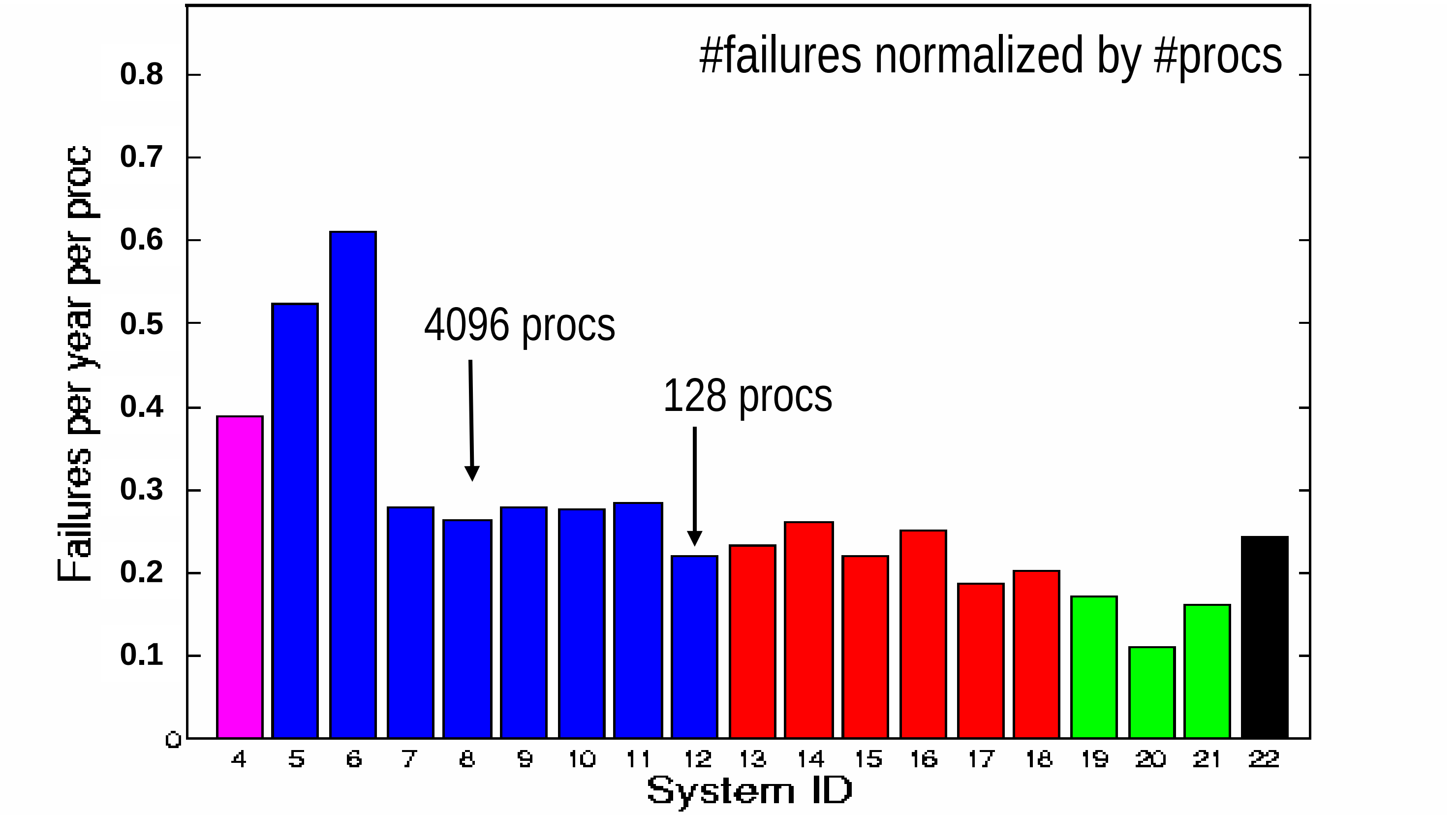


4096 procs

128 procs

- System failure rate varies from 20 – 1159 failures per year.
- Large variability even within systems of same HW type.

How does failure rate vary across systems?

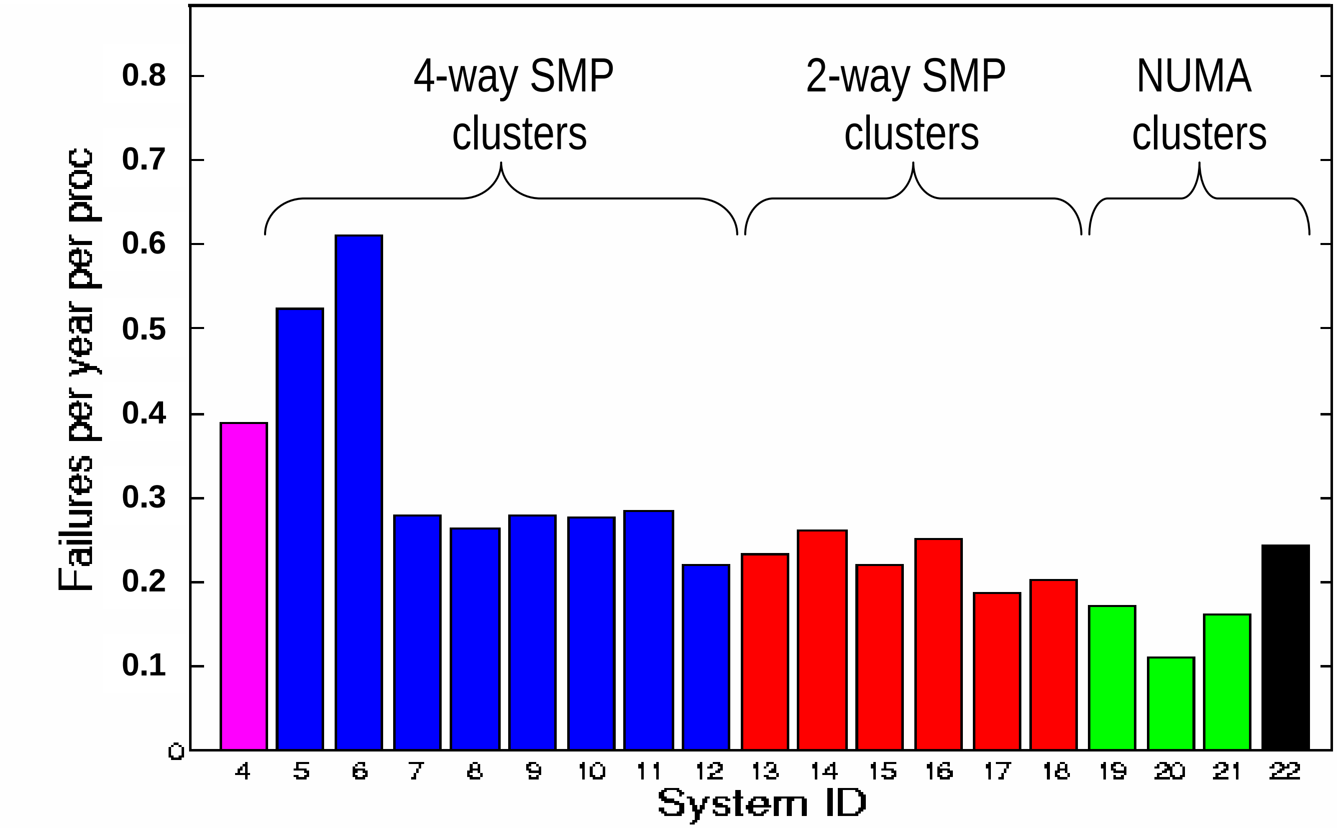


- Normalized failure rates are similar for system of same type, despite large size differences.

  => Failure rate grows ~linearly with system size.
- Similar even across systems of different type.

Carnegie Mellon
Parallel Data Laboratory

How does failure rate vary across systems?



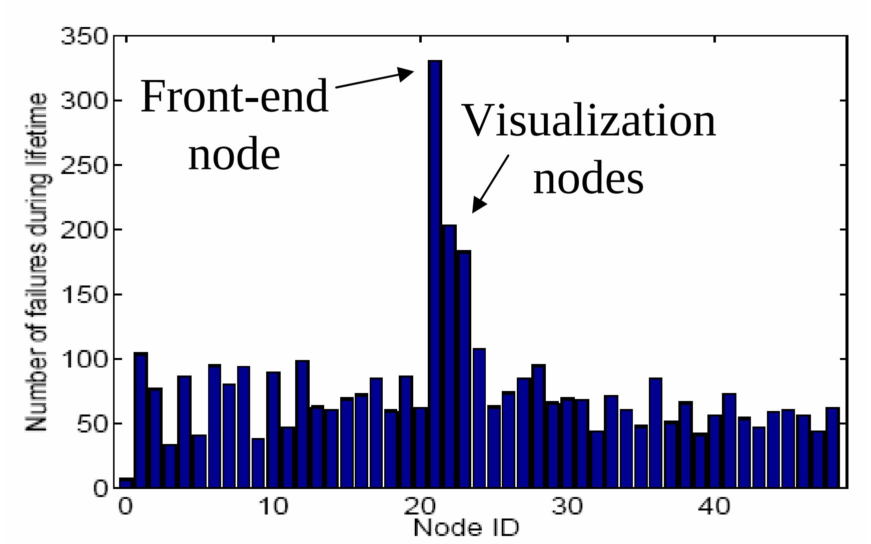
- Normalized failure rates are similar for system of same type, despite large size differences.

  => Failure rate grows ~linearly with system size.

- Similar even across systems of different type.

How does failure rate vary across nodes in a system?

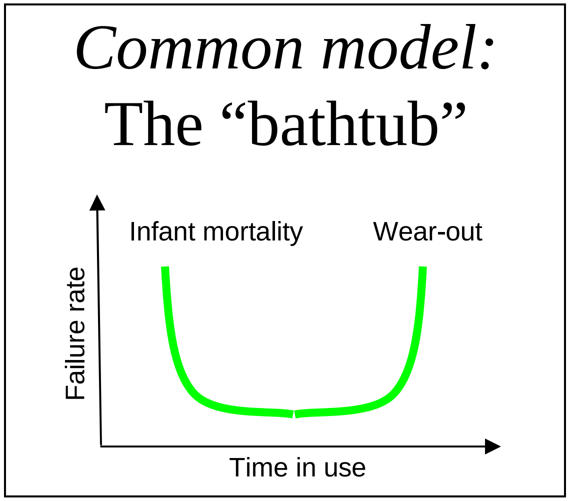
- *Common assumption:* Nodes see independent *Poisson processes* with equal mean.



- Large skew in distribution across nodes.

  => Front-end & visualization nodes have higher failure rate.
- Skew even in compute-only nodes.

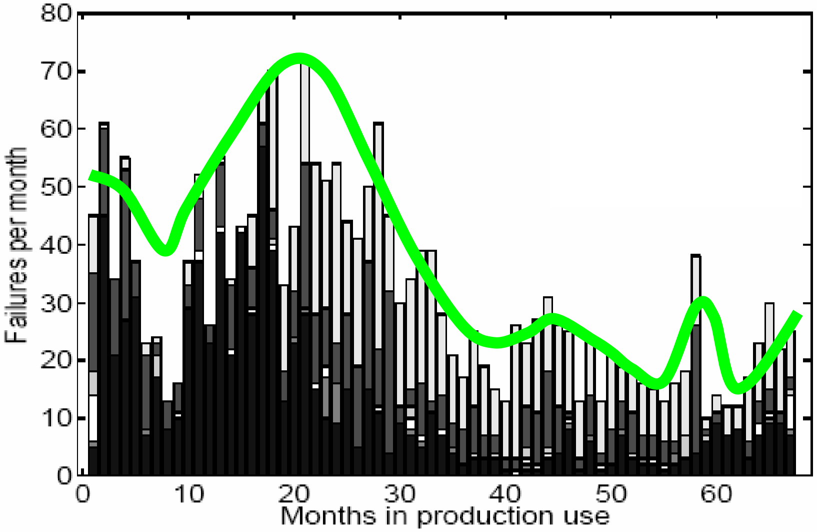
How does failure rate change over system lifetime?



Common model:

The "bathtub"

System 19

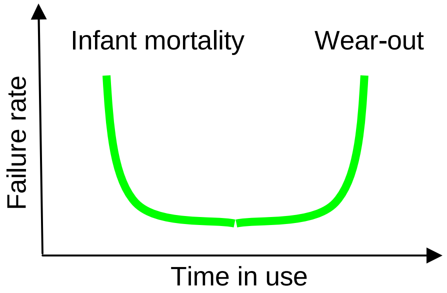


How does failure rate change over system lifetime?

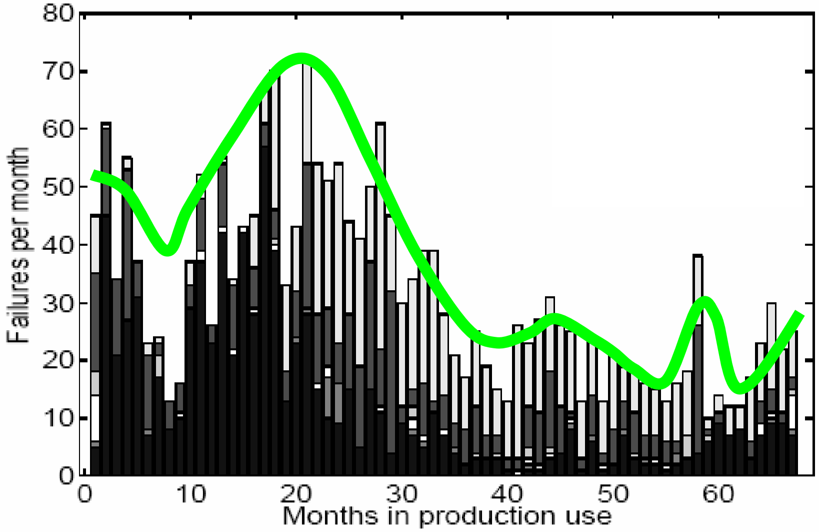
- Failure rate can look different from theoretical models such as the "bathtub".
- The shape of the curve varies greatly across systems.

Common model:

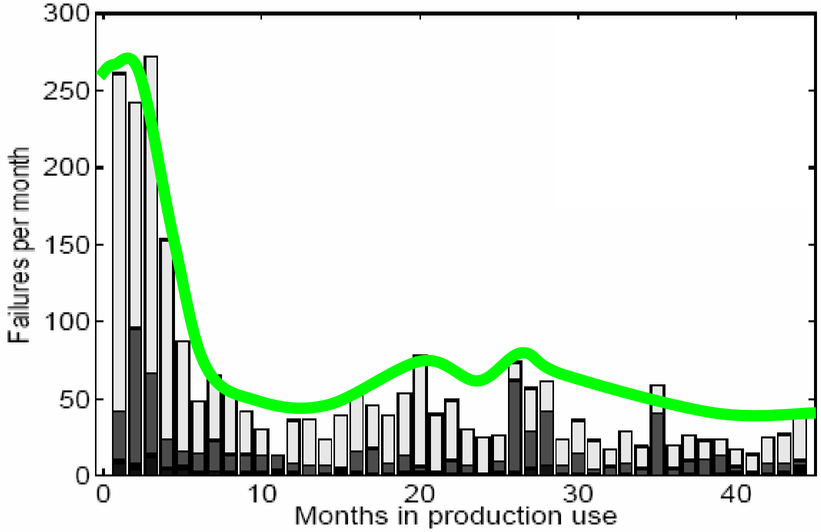
The "bathtub"



System 19

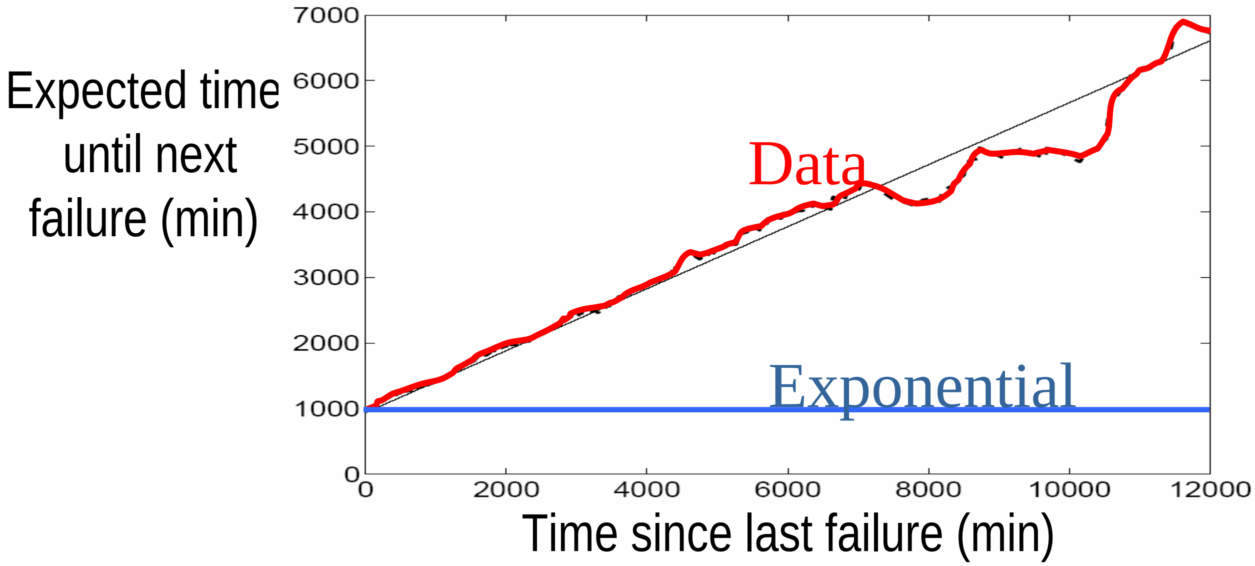


System 5



Statistical properties of time between failure

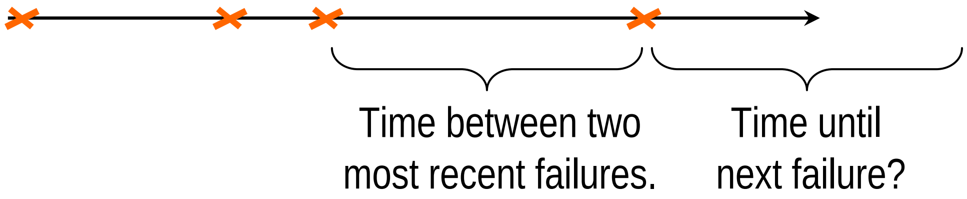
- *Common assumption:* Time between failure follows **exponential** distribution.
- LANL data differs from exponential:
  - Variability is higher (C^2 = 1.7--12).
  - Hazard rates are decreasing.

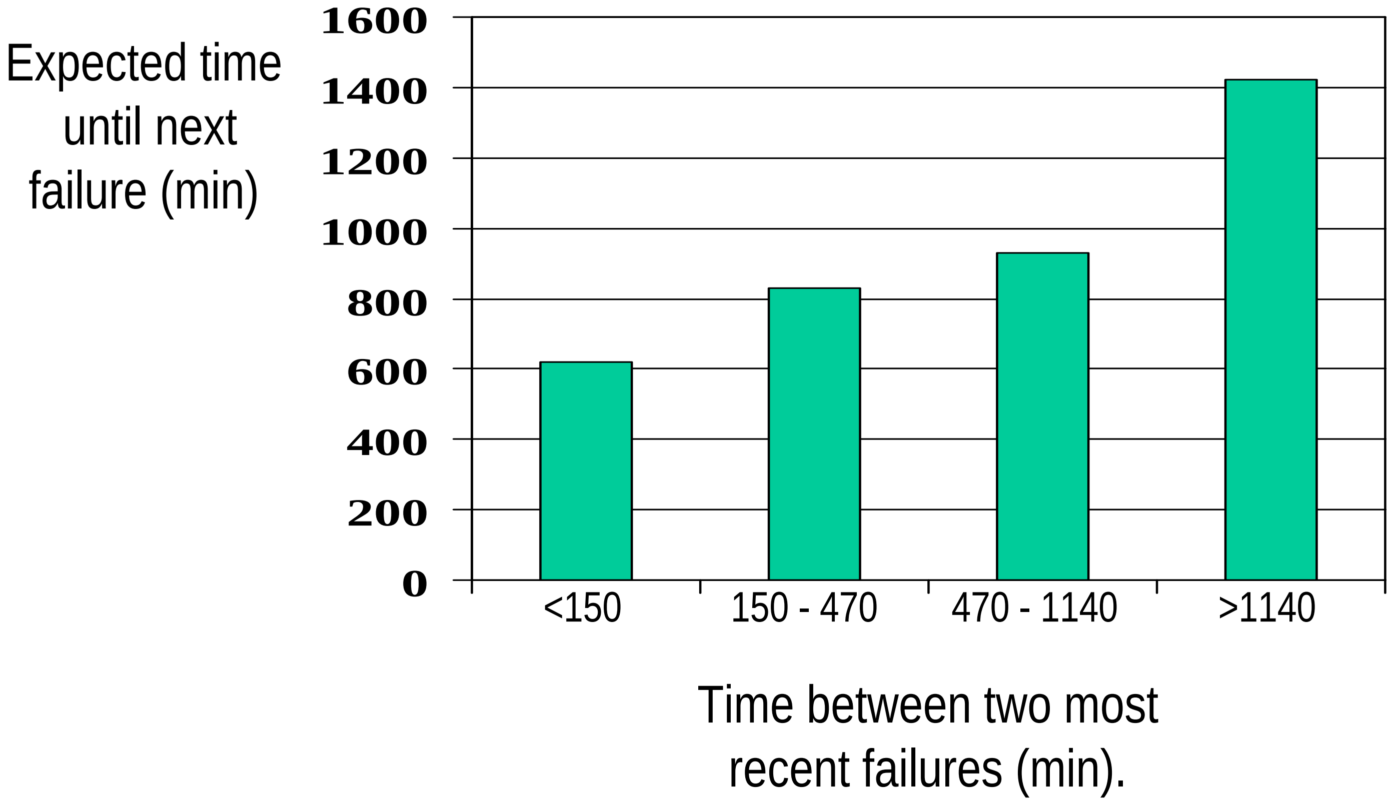


- Probability of failure decreases with time since last failure.
- Should checkpointing intervals really be fixed?

Statistical properties of time between failure

- *Common assumption:* Failures are independent.

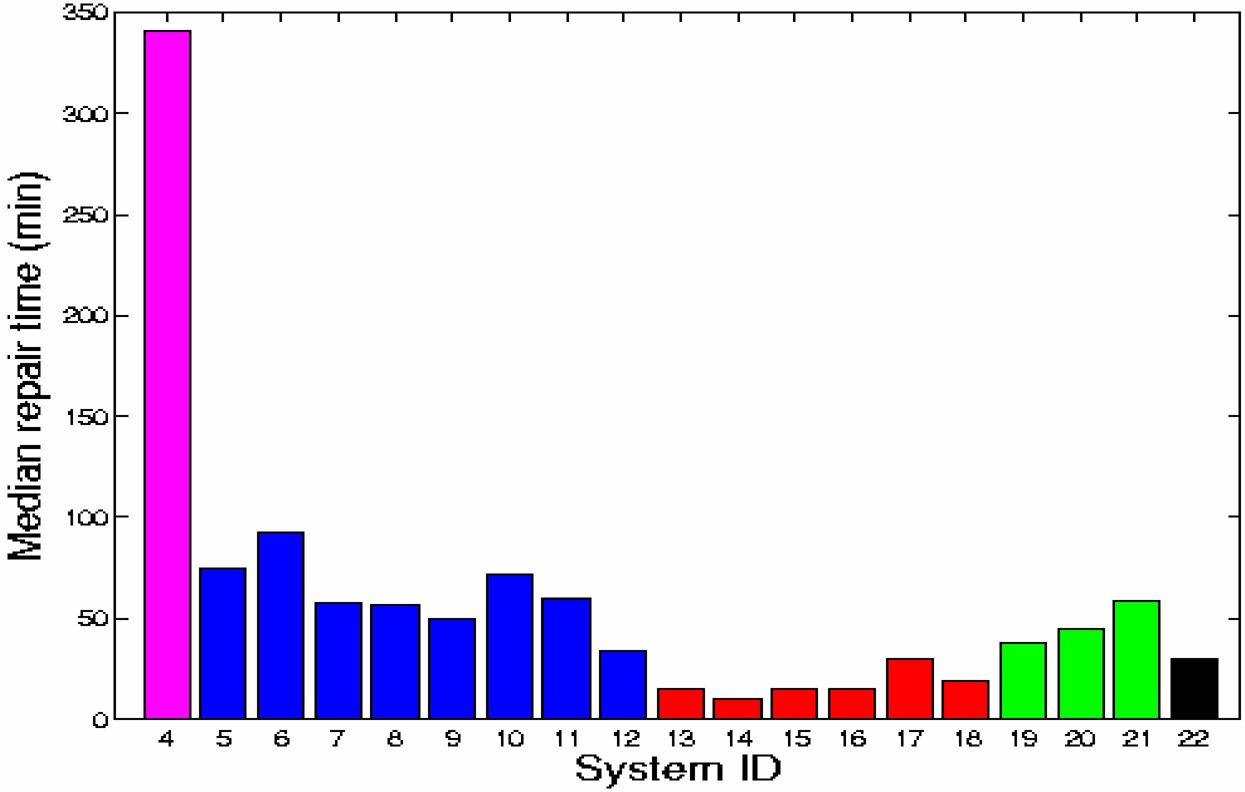




Outline

- What do **failure rates** (or time between failures) look like?
- What do **repair times** look like?
- What are the common **root causes** of failures?

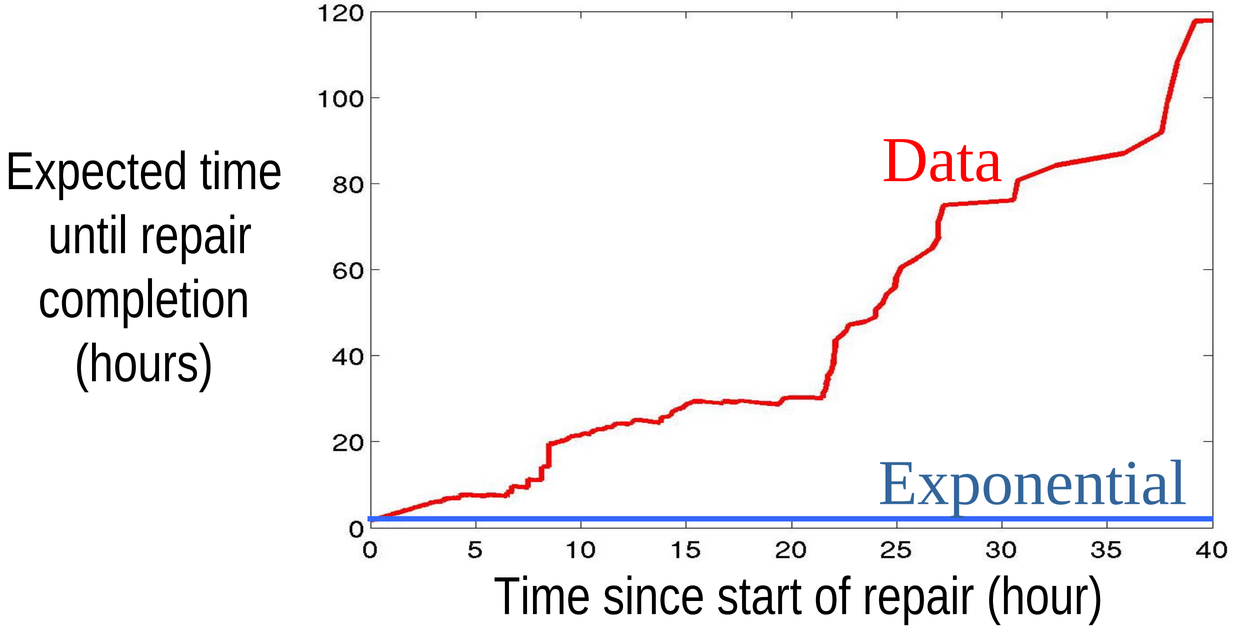
What do repair times look like?



- Median repair times vary from 10 – 350 min.
- Less variability within system of same HW type.
  - Little correlation with system size.

Statistical properties of repair times

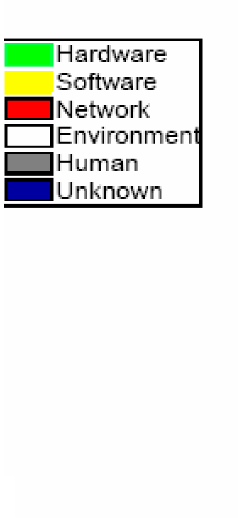
- *Common assumption:* Time to repair follows **exponential** distribution .
- LANL data differs from exponential:
    - Variability is higher (C^2 = 3 -- 200).
    - Hazard rates are decreasing.



Outline

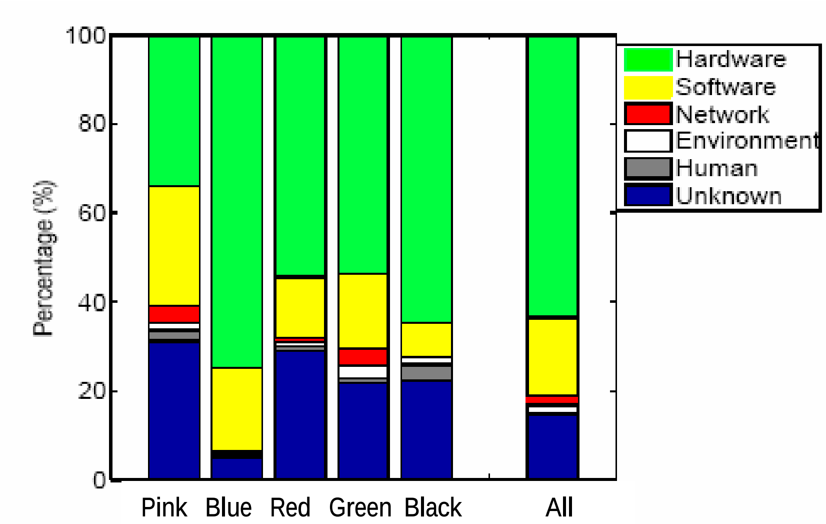
- What do **failure rates** (or time between failures) look like?
- What do **repair times** look like?
- What are the common **root causes** of failures?

What is the common root cause of failures?

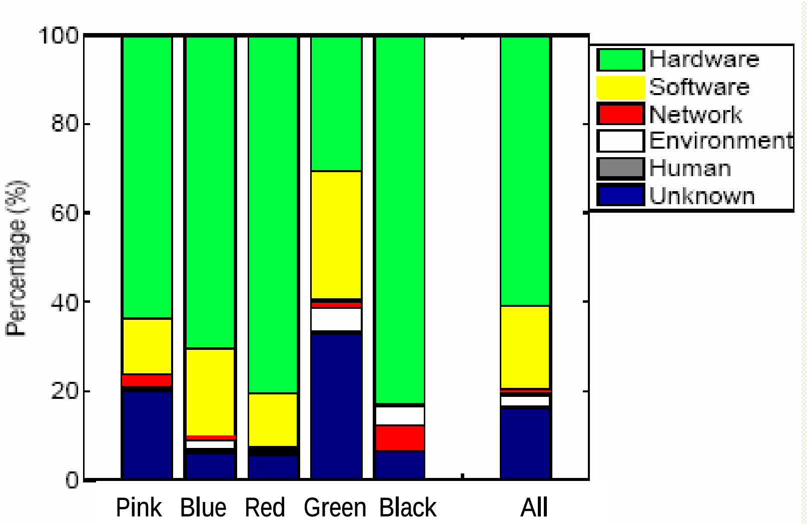


Relative frequency of root cause by system type.

What is the common root cause of failures?



Relative frequency of root cause by system type.



Fraction of total downtime caused by each root cause.

- Breakdown varies across systems.
- Hardware and software tend to be the most common root cause, and the largest contributors to system downtime.

Summary of data analysis

- Many common failure models are not realistic:
  - Failure rates and repair times are not exponential.
  - Failure rates are not i.i.d.
  - Failures are not evenly distributed over cluster nodes.
  - Failure rates over lifetime can look very different from bathtub.
- Failure rates
  - vary widely across systems
  - mostly depend on system size, not system type.
- Repair times
  - vary widely across systems
  - mostly depend on system type, not system size.
- Hardware and software related failures dominate in HPC environment.

Long-term research goals

- Create public failure data repository.
  - Collect data from diverse set of sites.
  - Add other types of data
    - Error logs.
    - Utilization and workload data.
    - Sensor data.
    - Storage data.
- Best practices for data collection
  - How much data is enough?

Long-term research goals

- Analysis of data
  - Statistical properties.
    - Distributions
    - Correlations
  - How do you deal with imperfect data?
  - Compare with commonly made assumptions.

- More realistic performance evaluation
  - Data-driven dependability *benchmarking*.
  - What are the right *failure models* for dependability simulation, analysis and experiments?
    - As *realistic* as possible.
    - Yet *simple* …

Will require collaborative effort of researchers in

- Statistics.
- Data mining.
- Computer systems.
- Performance modeling.

Long-term research goals

- Exploit data for building better systems
  - Can we exploit statistical properties (e.g. decreasing hazard rates) to improve fault tolerance?
  - Proactive fault tolerance?
  - Automated problem diagnosis?

Will require collaborative effort of researchers in
- Computer systems.
- Data mining.
- Machine learning.

Thank you!
Questions?