Learning to live with our failures

Bianca Schroeder

Joint work with Garth Gibson

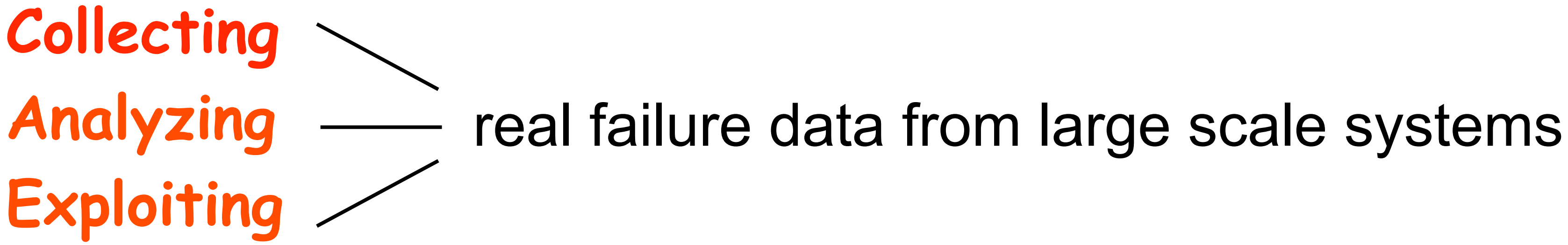
PARALLEL DATA LABORATORY

Carnegie Mellon University

Petascale computing and reliability

- Component failure will be the norm.
- Dealing with it requires understanding of what failures look like in real, large-scale systems.

One goal of PDSI:



Collecting

Analyzing

Exploiting

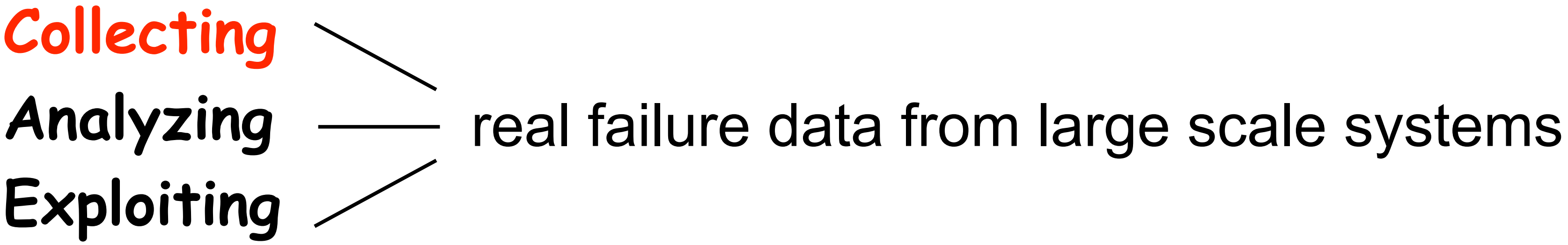
real failure data from large scale systems

Goal of this talk:

1. **Status report** – where are we now?
   - DSN’06
   - FAST’07
2. **Your feedback** – where should we go next?

Your opinion counts ….

Outline:



Collecting

Analyzing

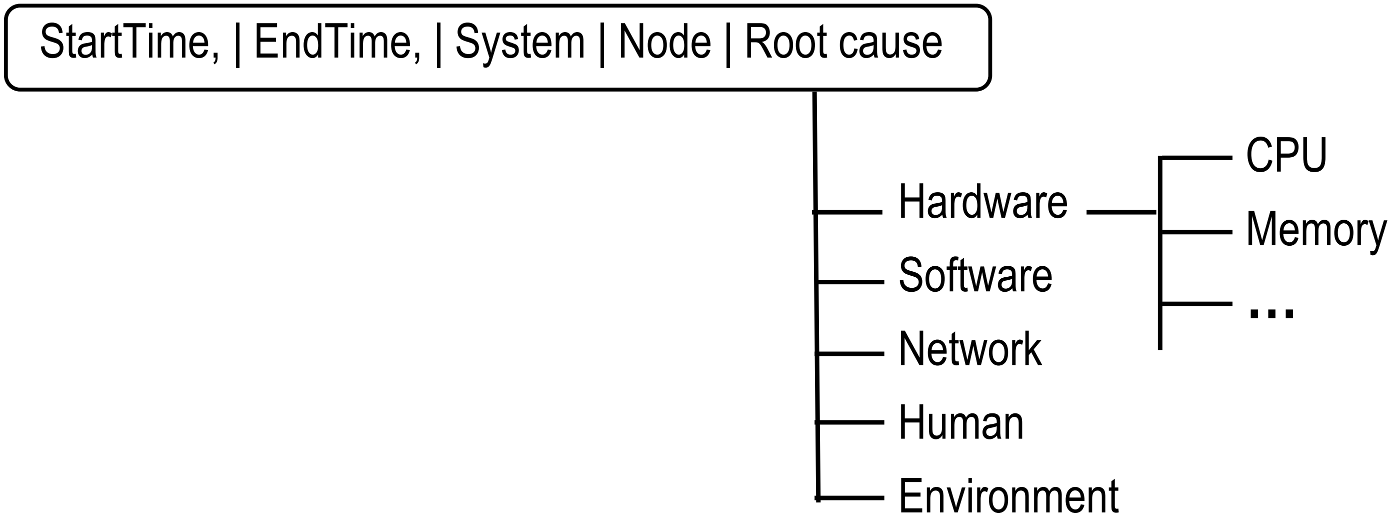
Exploiting

real failure data from large scale systems

The computer failure data repository

- To be supported by the Usenix association.
- So far, data from 26 large systems at 3 sites.

Los Alamos NATIONAL LABORATORY EST.1943	22 HPC clusters	5000 nodes	9 years	Node outages Usage data Event logs



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

Hardware

Software

Network

Human

Environment

CPU

Memory

...

The computer failure data repository

- To be supported by the Usenix association.
- So far, data from 26 large systems at 3 sites.

Los Alamos NATIONAL LABORATORY EST.1943	22 HPC clusters	5000 nodes	9 years	Node outages Usage data Event logs
Pittsburgh Supercomputing Center	1 HPC cluster	765 nodes	5 years	Hardware/ disk drive replacements
Internet services X	3 storage clusters	70,000 disks	1 mth – 5 yrs	
More coming soon …				

Your opinion counts ….

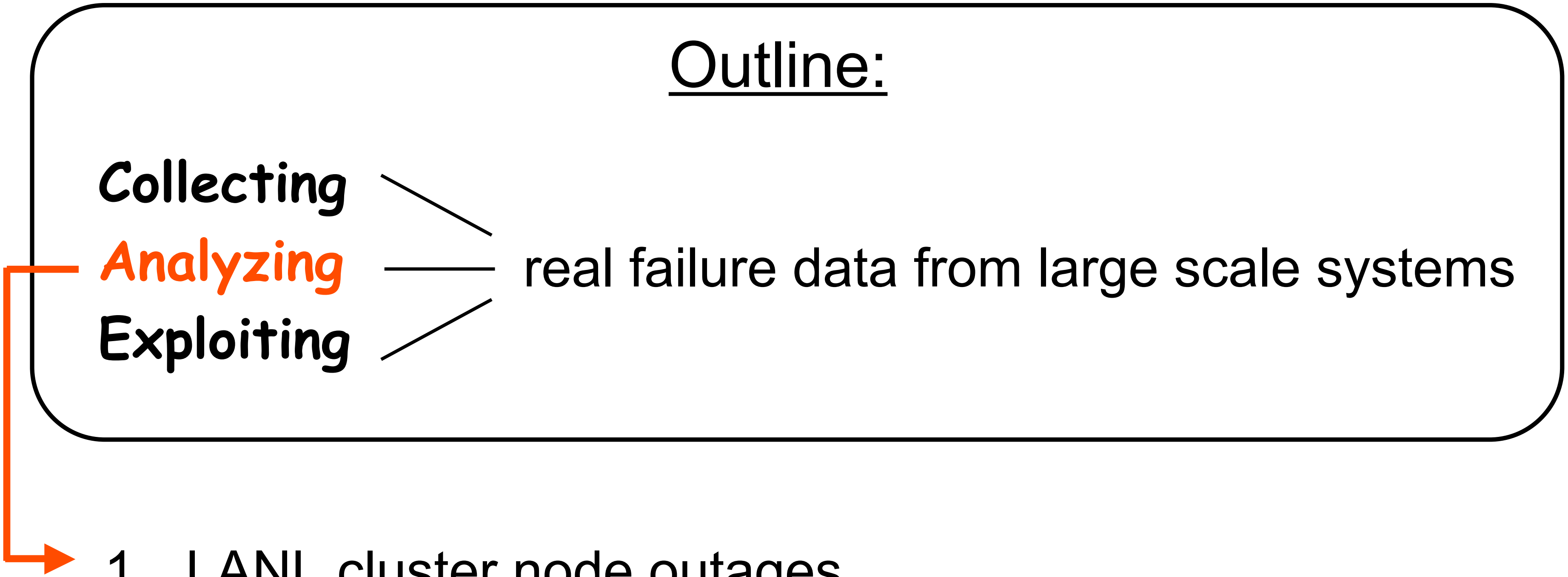
□ What else to gather?

- Other systems?
- Other types of data?
- Who might be willing to share?

□ Ideas on anonymizing data?

□ Ideas on automatically parsing data?

□ Best practices for data collection?



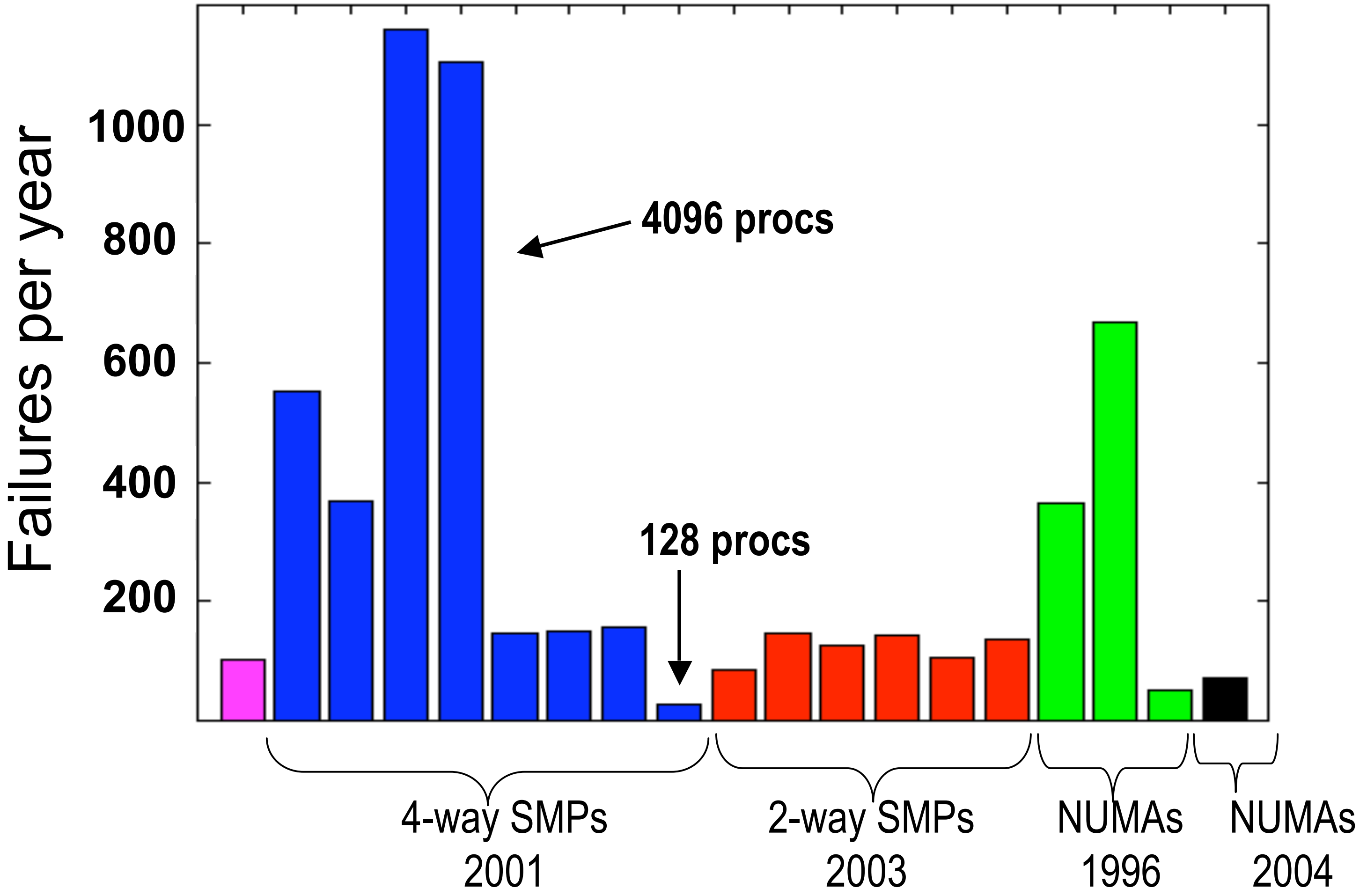
Outline:

Collecting
Analyzing
Exploiting

real failure data from large scale systems

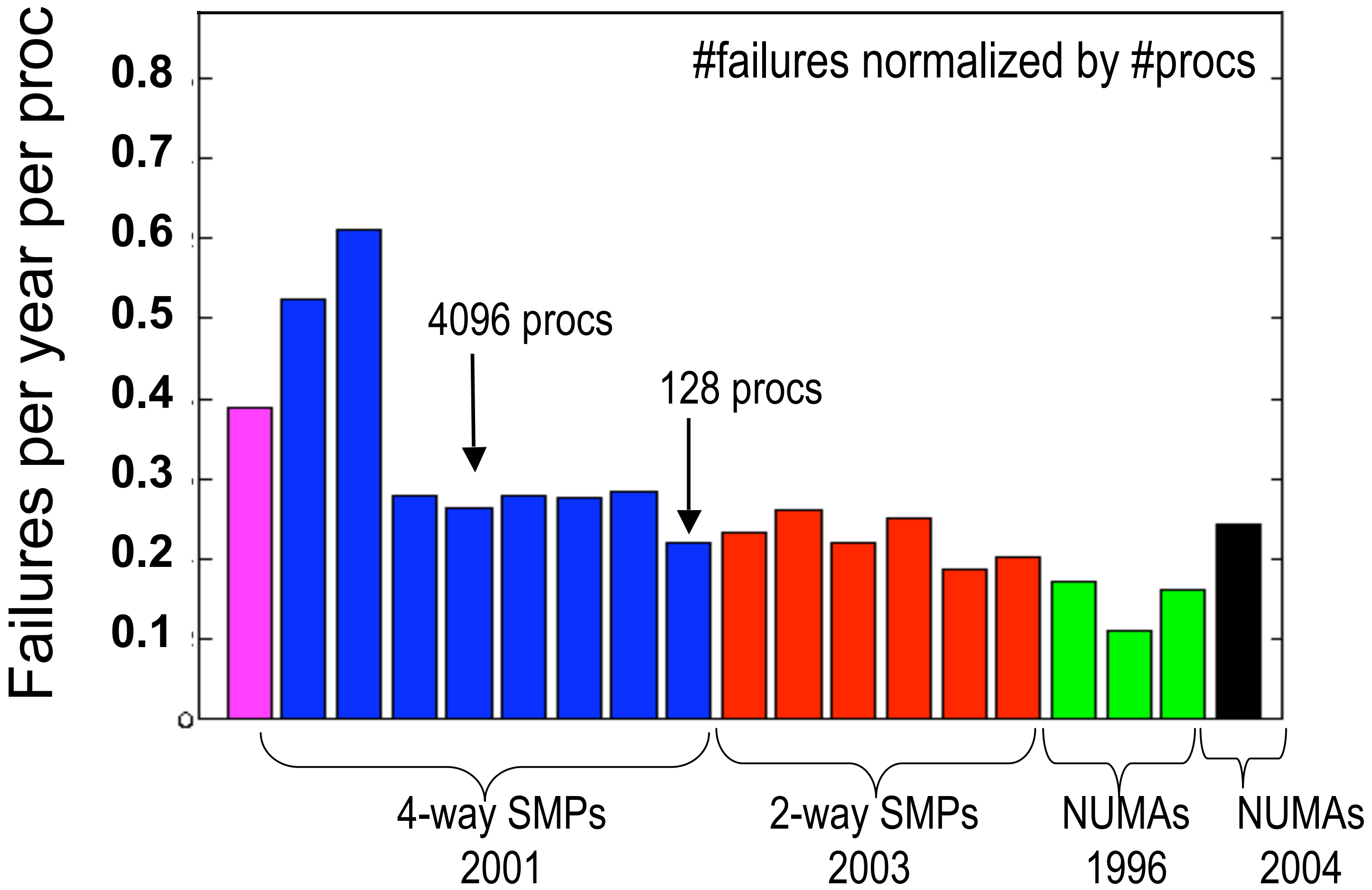
1. LANL cluster node outages
2. Storage failures
3. Statistical properties of failures

What do failure rates look like?



- Large variability -- even within systems of same HW type.

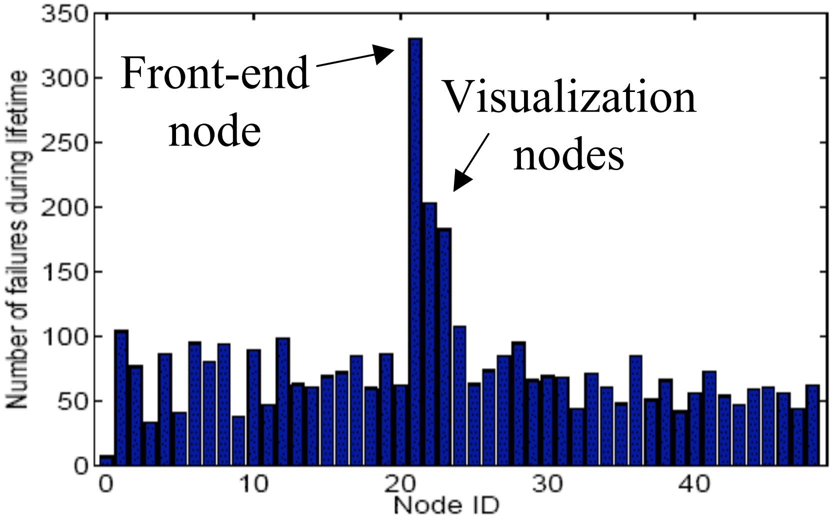
How does failure rate vary across systems?



- Normalized failure rates similar, despite size differences
  => Failure rate grows ~linearly with system size.
- Similar even across systems across different type & age.

How does failure rate vary across nodes in a system?

- *Common assumption:* All nodes are equally likely to fail.



- Large skew in distribution across nodes.

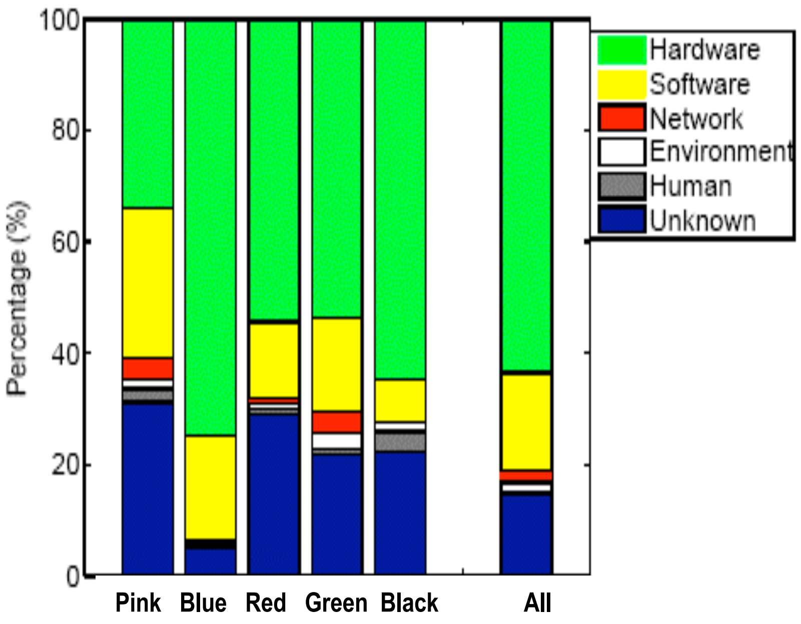
  => Front-end & visualization nodes have higher failure rate.
- Skew even in compute-only nodes.

What is the common root cause of failures?

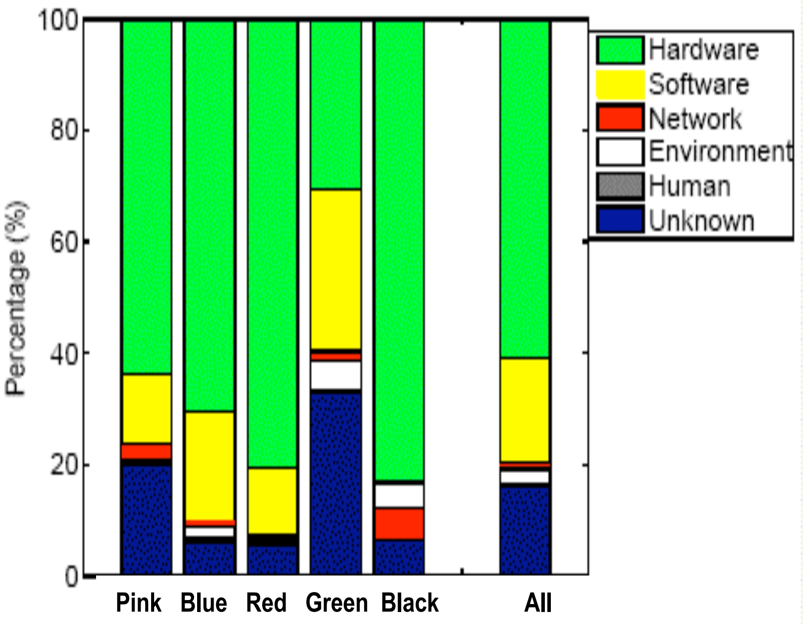
Pink Blue Red Green Black All

Relative frequency of root cause by system type.

What is the common root cause of failures?



Relative frequency of root cause by system type.



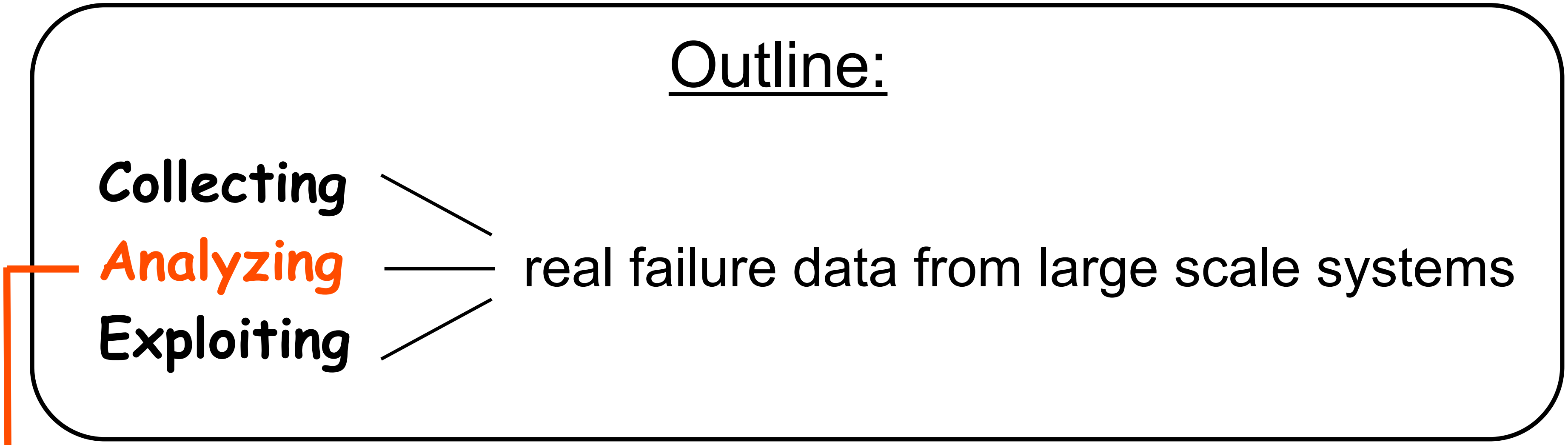
Fraction of total repair time caused by each root cause.

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

Your opinion counts ….

❑ What else to explore in LANL data?

- Workload data
- Event data
- …. what else?



Outline:

Collecting
Analyzing
Exploiting

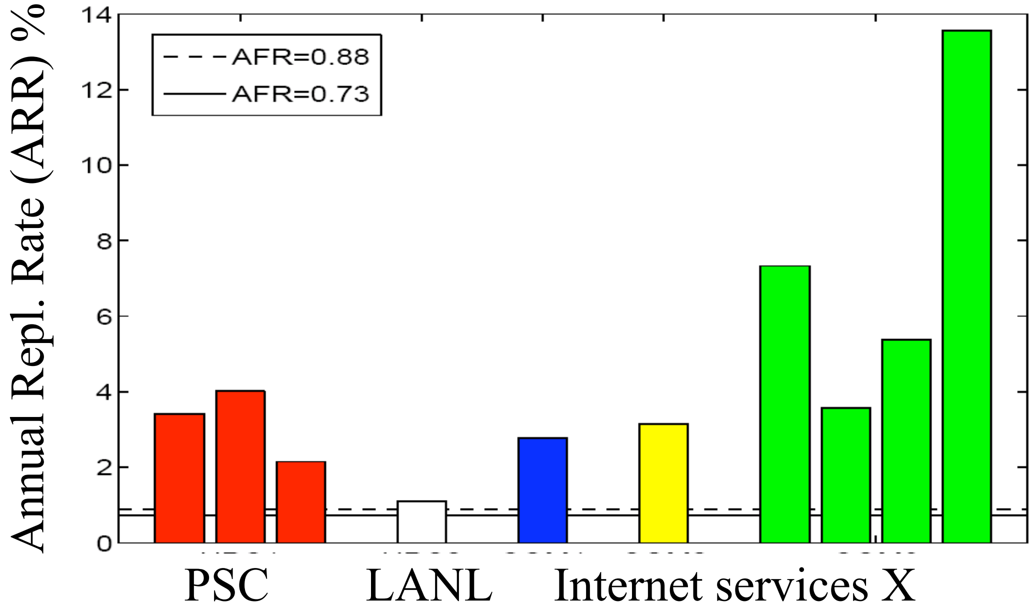
real failure data from large scale systems

1. LANL cluster node outages
2. Storage failures
3. Statistical properties of failures

Annual replacement rate (ARR) in the field

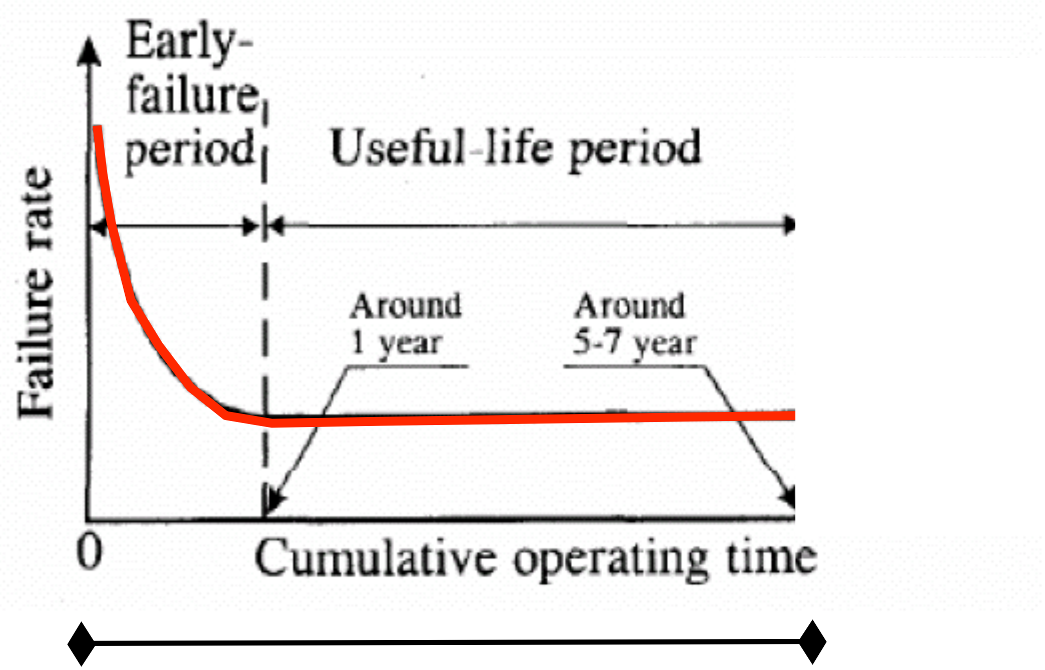
- Datasheet MTTF is 1,000,000 to 1,200,000 hours for disks in data.

=> Expected annual failure rate (AFR) is 0.73 - 0.88 %.



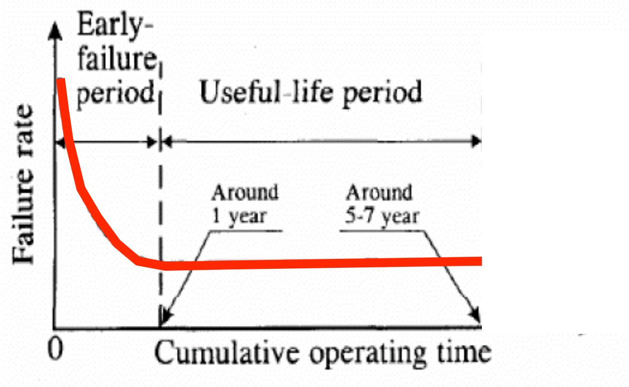
- Field replacement is a fairly different process from what one might predict based on datasheet MTTF.

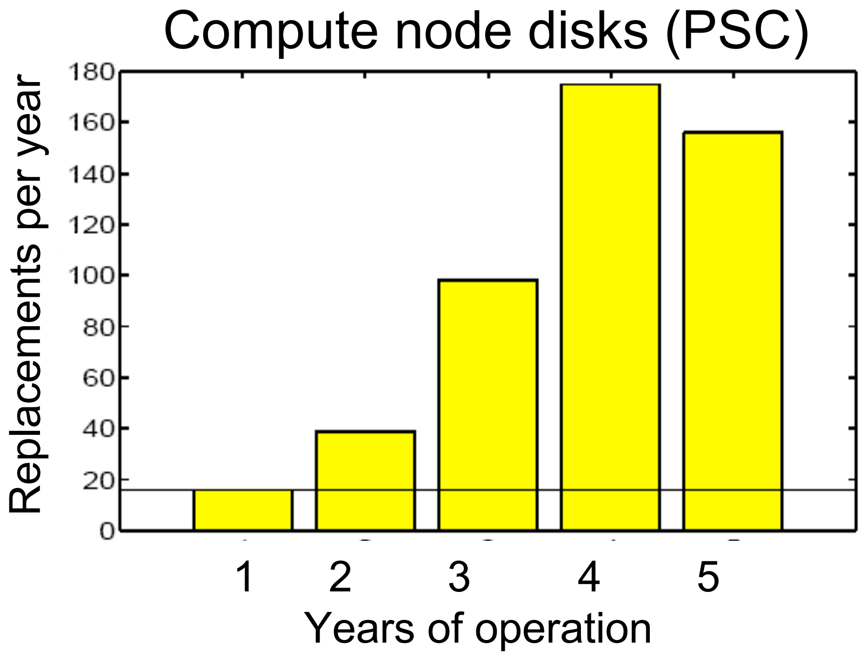
Failures as a function of age - model

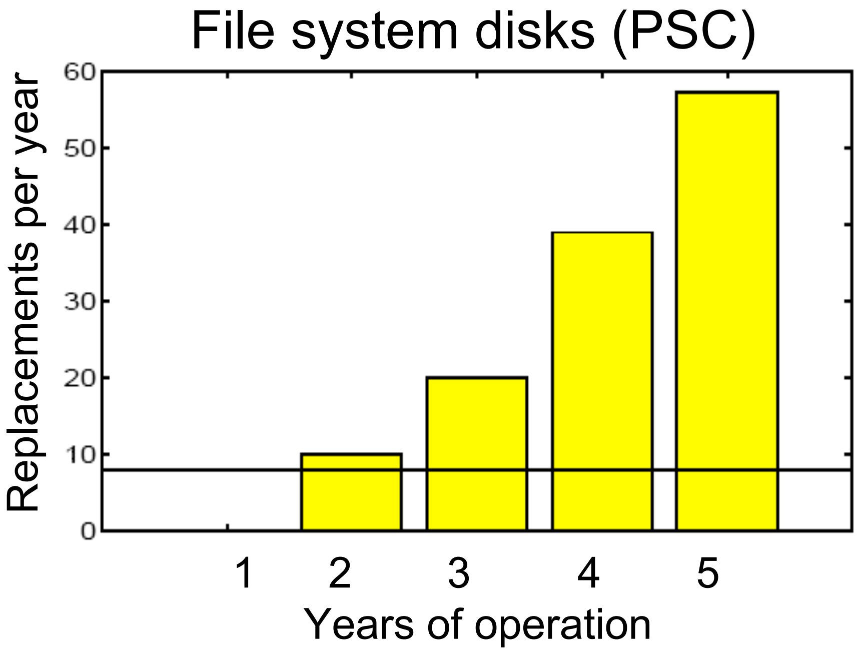


Nominal lifetime – 5 years

Replacements as a function of age in the field







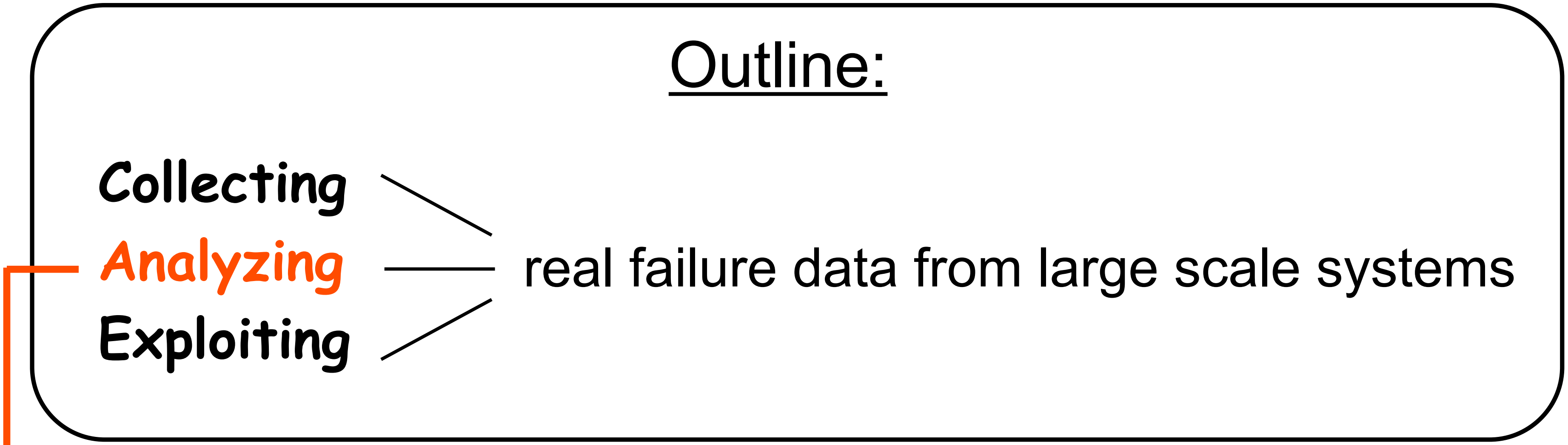
Your opinion counts ….

❑ What else to explore in storage data?

❑ What other data to gather?

- Usage data
- Temperature data
- SMART data
- Media errors

❑ Where can we get other/more data?



Outline:

Collecting
Analyzing
Exploiting

real failure data from large scale systems

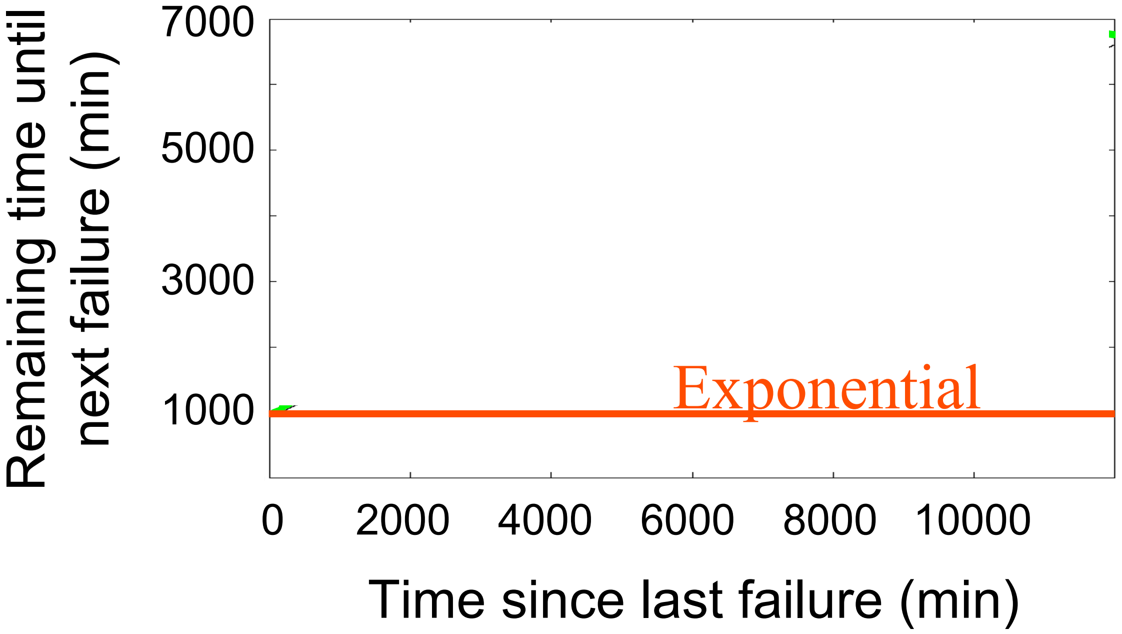
1. LANL cluster node outages
2. Storage failures
3. Statistical properties of failures

Statistical properties of failures

- Common assumption:
  - Time between failures is exponentially distributed.
  - Failures are independent.

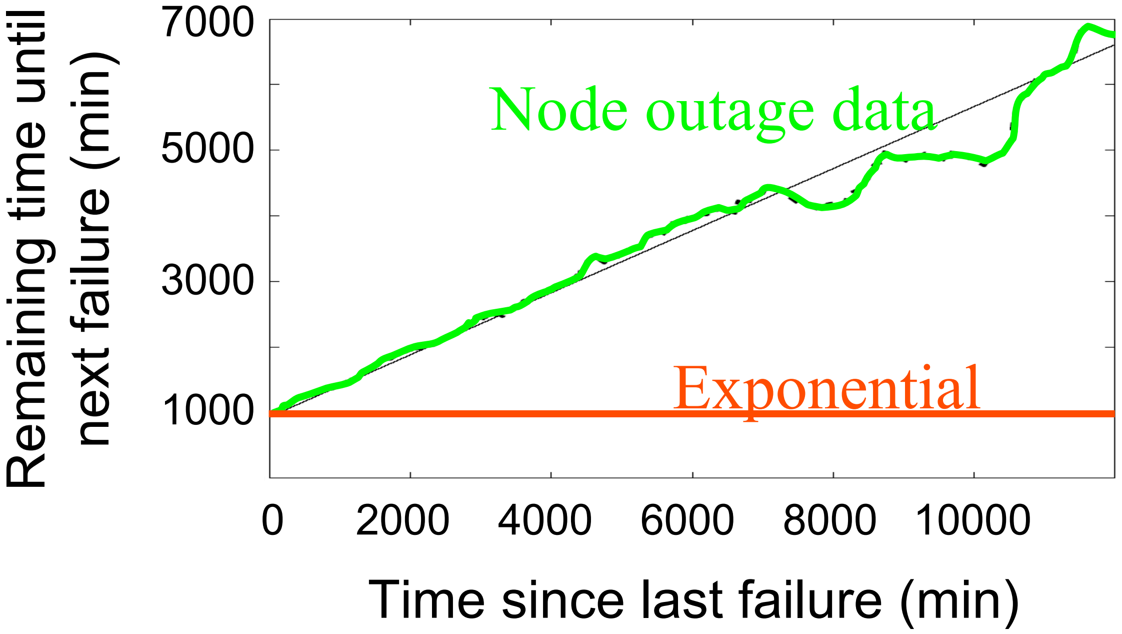
Statistical properties of time between failure

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



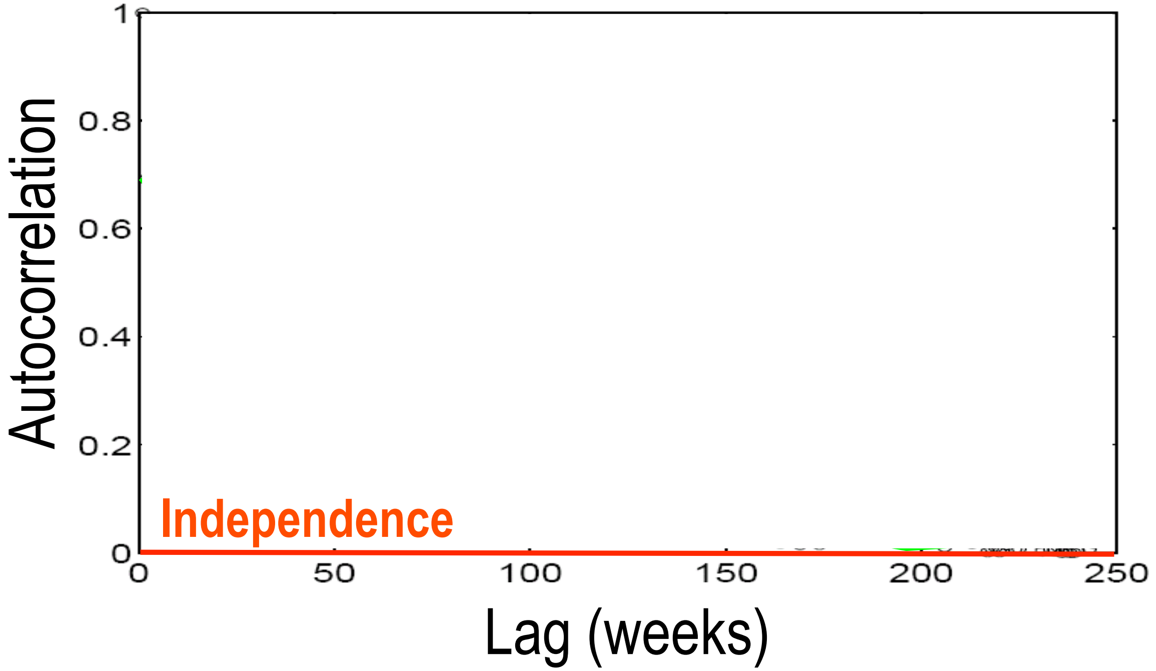
Statistical properties of time between failure

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



Statistical properties of time between failure

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



Your opinion counts ….

- What other properties to look at?

- What's relevant for your application?

Outline:

Collecting

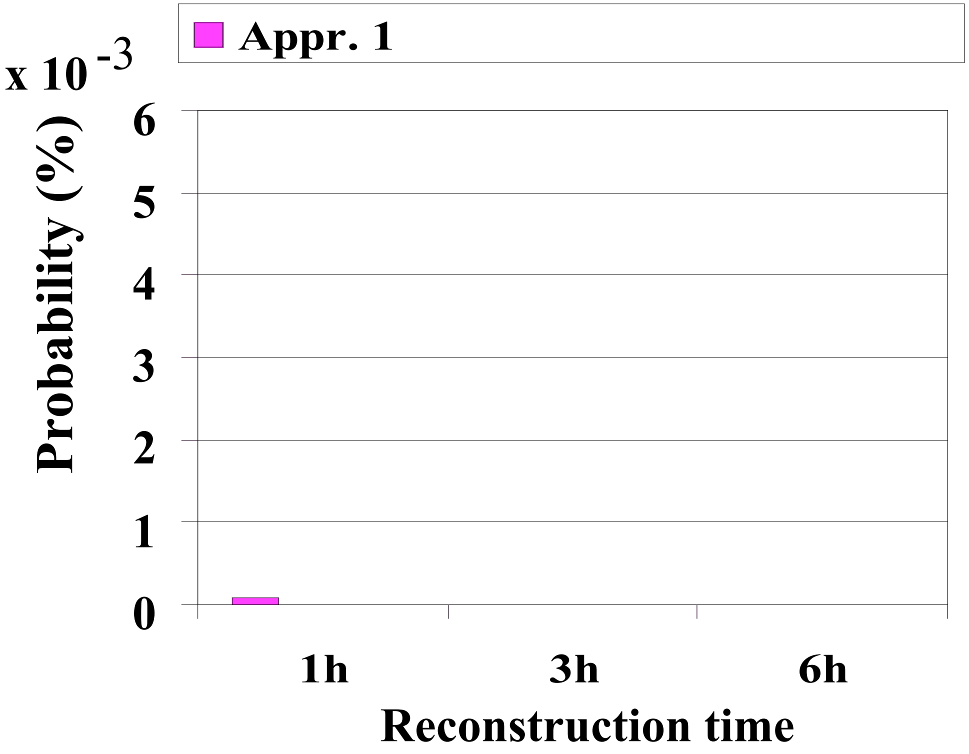
Analyzing

Exploiting

real failure data from large scale systems

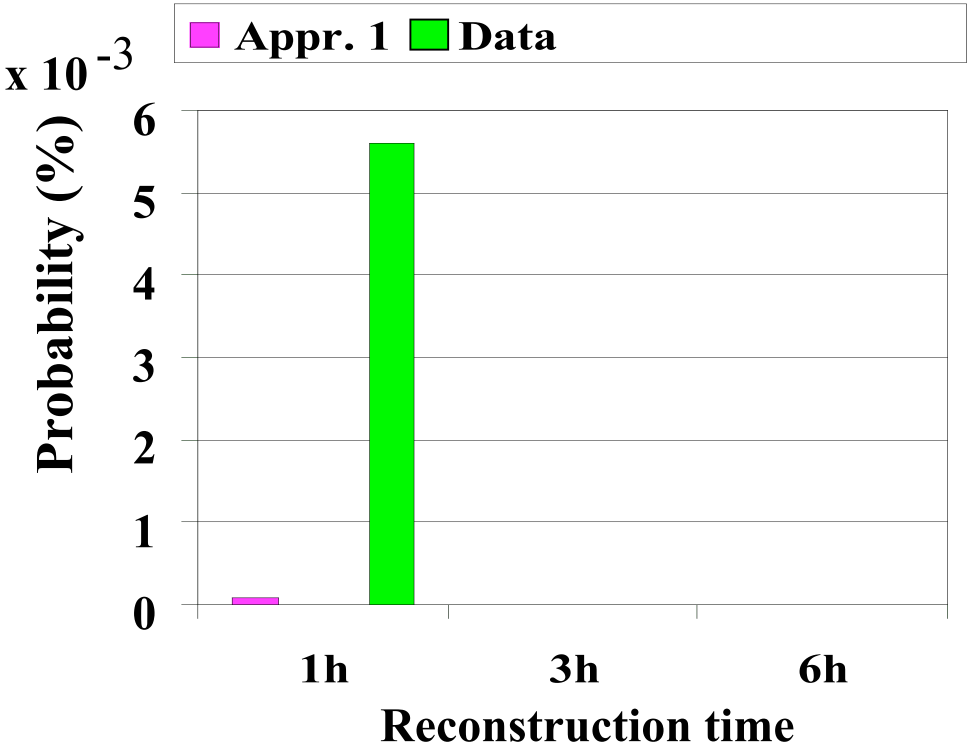
Probability of a RAID failure

- Depends on probability of second failure during reconstruction.
- <u>Approach 1:</u> Use datasheet MTTF and exponential distribution.



Probability of a RAID failure

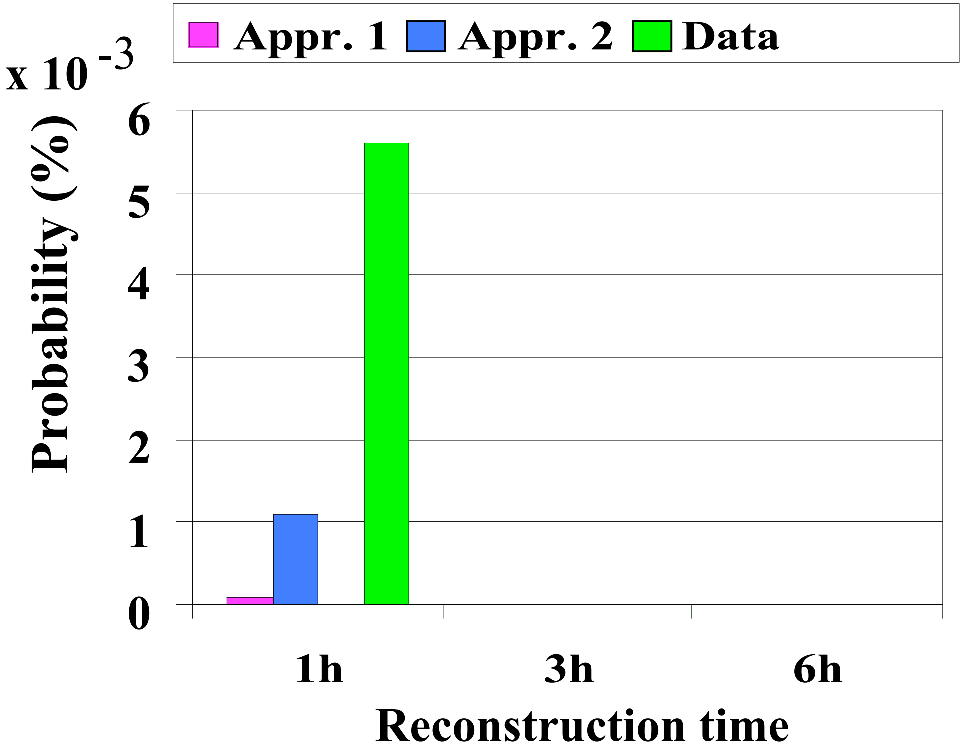
- Depends on probability of second failure during reconstruction.
- Approach 1: Use datasheet MTTF and exponential distribution.



Probability of a RAID failure

- Depends on probability of second failure during reconstruction.

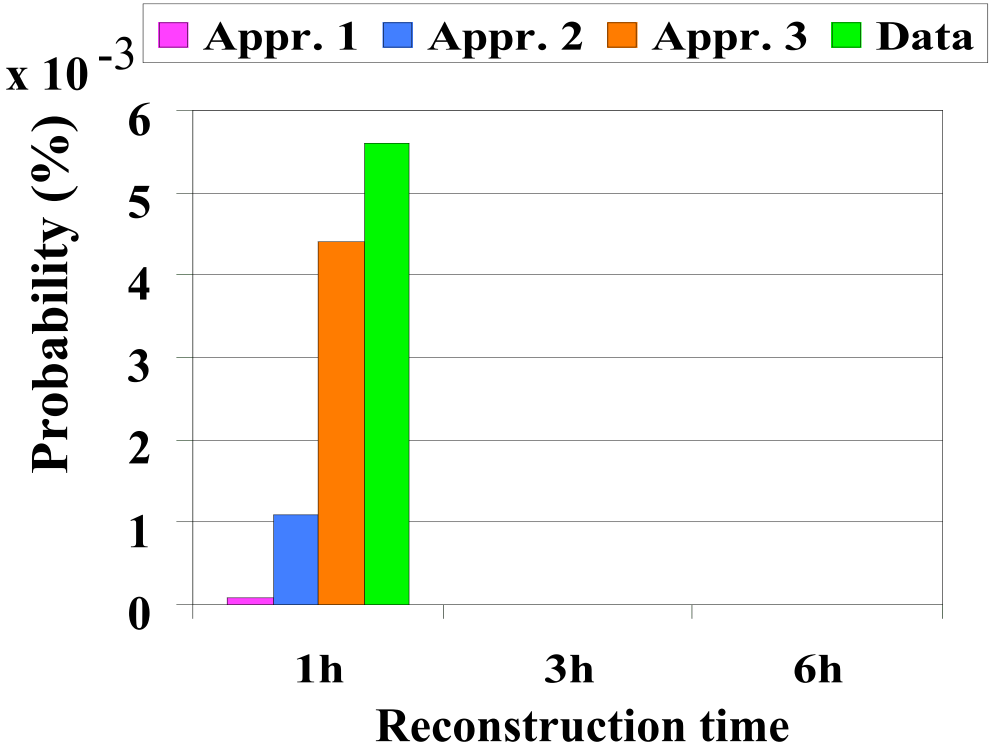
- Approach 1: Use datasheet MTTF and exponential distribution.
- Approach 2: Use **measured** MTTF and exponential distribution.



Probability of a RAID failure

- Depends on probability of second failure during reconstruction.

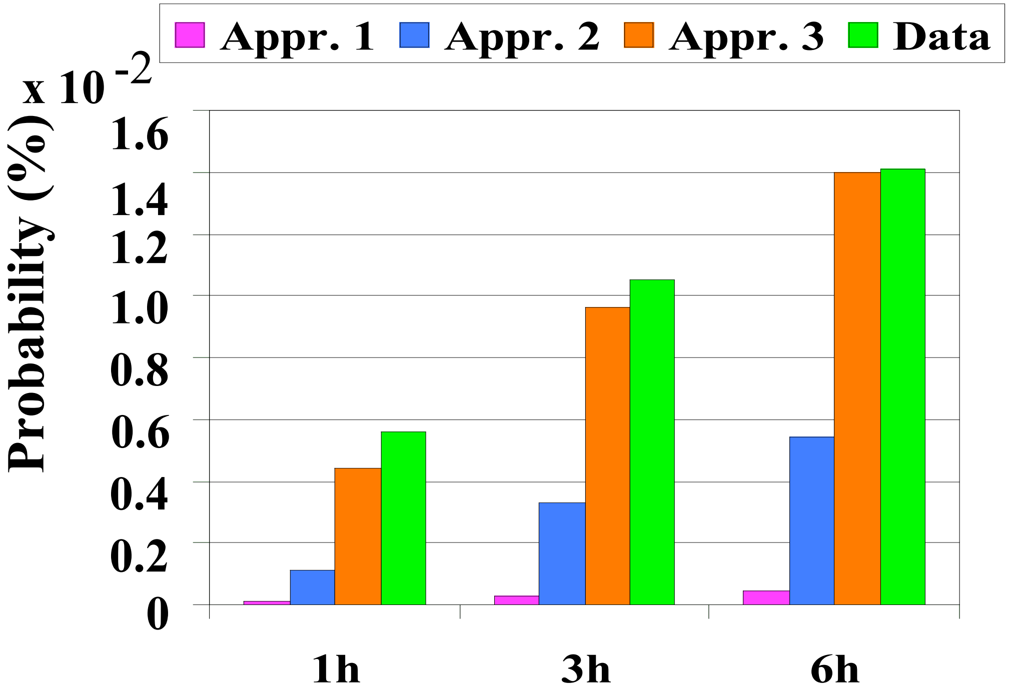
- Approach 1: Use datasheet MTTF and exponential distribution.
- Approach 2: Use **measured** MTTF and exponential distribution.
- Approach 3: Use Weibull distribution fit to data.



Probability of a RAID failure

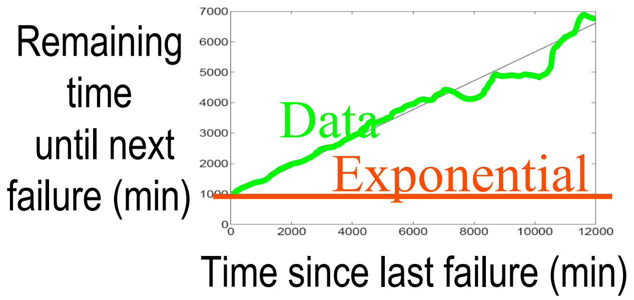
- Depends on probability of second failure during reconstruction.

- Approach 1: Use datasheet MTTF and exponential distribution.
- Approach 2: Use **measured** MTTF and exponential distribution.
- Approach 3: Use Weibull distribution fit to data.



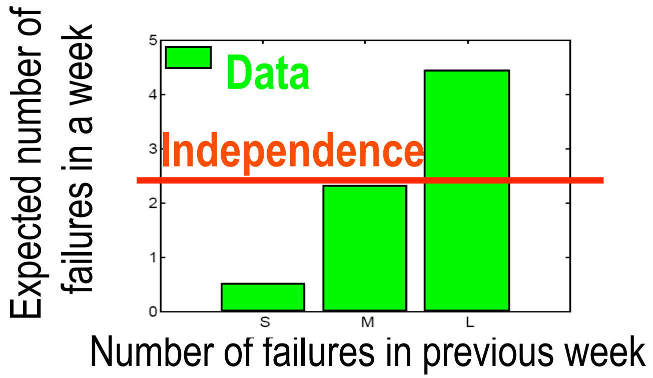
Non-exponential failures in HPC systems

- Fault-tolerance by check-point restart.
- Performance depends on choice of (fixed) checkpoint interval.
  - Too short: a lot of overhead writing back checkpoints.
  - Too long: a lot of lost work in case of failure.
- Use statistical properties to optimize checkpoint interval?



Idea:
Adapt checkpoint interval
based on past failure behavior.

Prelim. Results:
Up to 7-60% savings in overheads.



Your opinion counts ….

❑ What other applications to look at?

❑ Where do failure properties matter?

❑ What failure properties matter?

❑ Where else can we make use of failure data?

Conclusion

- Many common assumptions about failures are not realistic, based on our data analysis.
- Motivation for a lot of future work.
  - Create public failure data repository.
    - Data from large variety of systems.
  - Build more realistic models for system evaluation.
  - Exploit data for building better systems
    - Can we exploit statistical properties?
    - Automate & get proactive.
      - Automated problem diagnosis?
      - Failure signatures?
      - Proactive fault tolerance?

Collecting data

□ What else to gather?

□ Ideas on anonymizing data?

□ Ideas on automatically parsing data?

□ Best practices for data collection?

Analyzing data

□ What other properties to look at?

□ What's relevant for your application?

Exploiting data

□ What other applications to look at?

□ Where do failure properties matter?

□ Where else can we make use of failure data?