Reliability in Supercomputing - Trends and Challenges

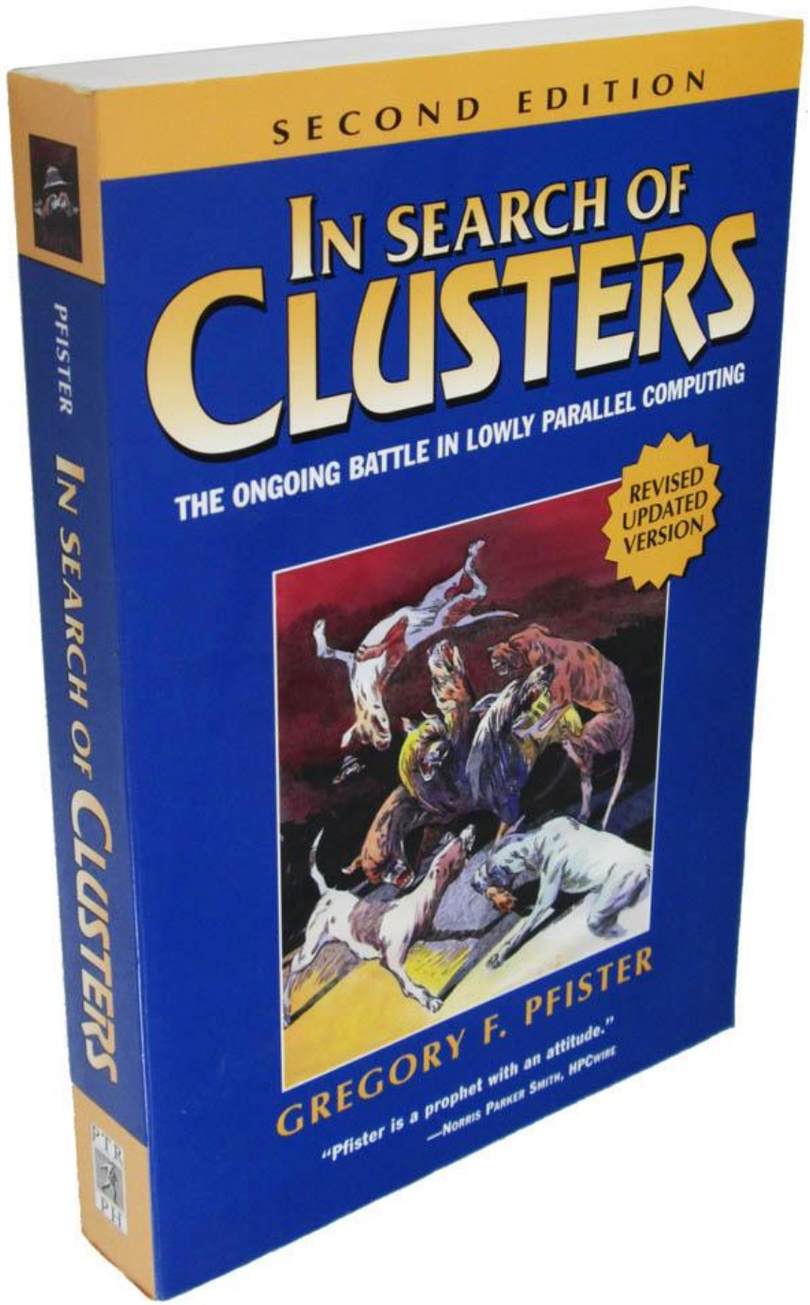
Peter Tröger (@ptroeger)

Beuth University of Applied Sciences
Berlin

GridKa Summer School

August 27th 2018

Supercomputing



- There are three ways of doing anything faster:
  - Work harder
  - Work smarter
  - Get help
- Next station: Exascale.

Exascale Computing

[top500.org]

Exascale Computing

Fujitsu Reveals Details of Processor That Will Power Post-K Supercomputer

Michael Feldman | August 22, 2018 16:00 CEST

Fujitsu has announced the specifications for A64FX, an Arm CPU that will power Japan's first exascale supercomputer. The system, known as Post-K, is scheduled to begin operation in 2021.

Read more

[top500.org]

- *„Using the conservative 2.7 teraflops/chip estimate, it will take **more than 370 thousand of these chips** to get to a peak exaflop …“*

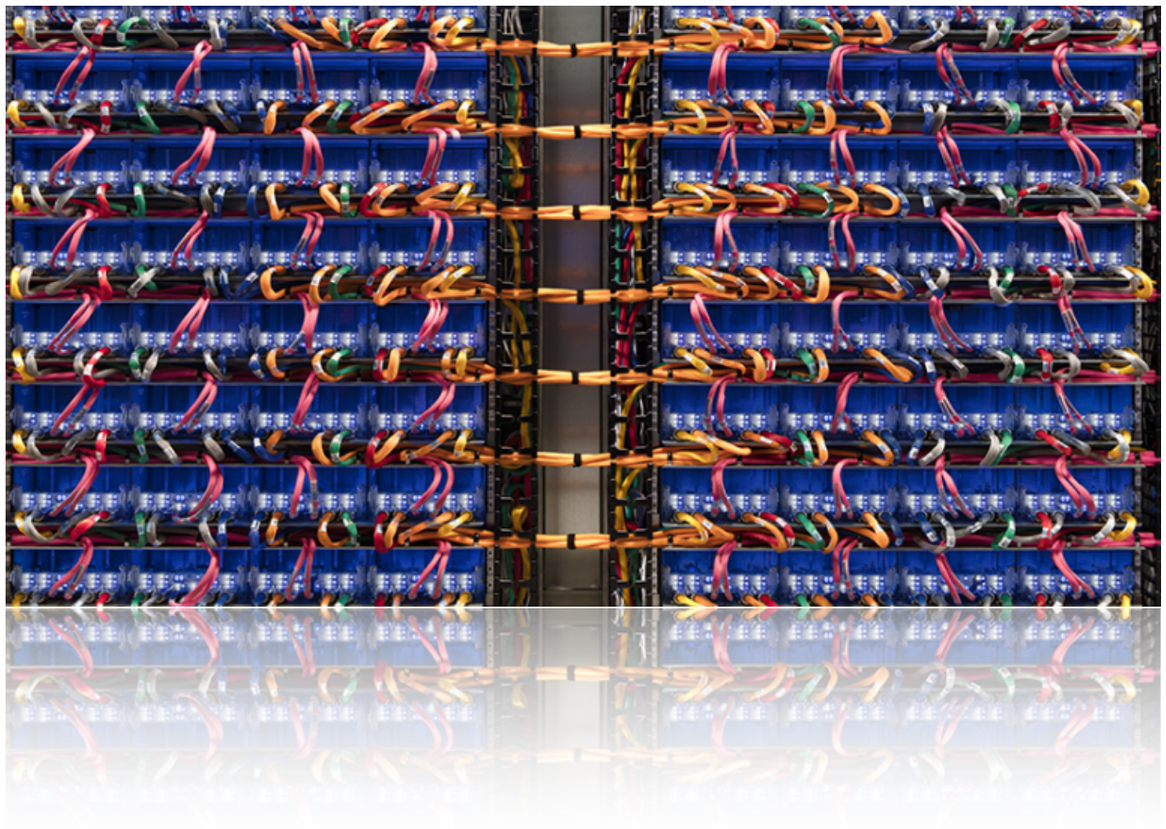
- *„… the system would need **more than 1,000 such racks** in the final exascale machine.“*

Exascale Computing

- Millions of compute cores
- Billions of concurrent activities
- Faults in the order of minutes
- Checkpointing takes hours
- Silent data corruption everywhere
- Software faults everywhere
- Redundancy limited by costs and power wall (20MW)

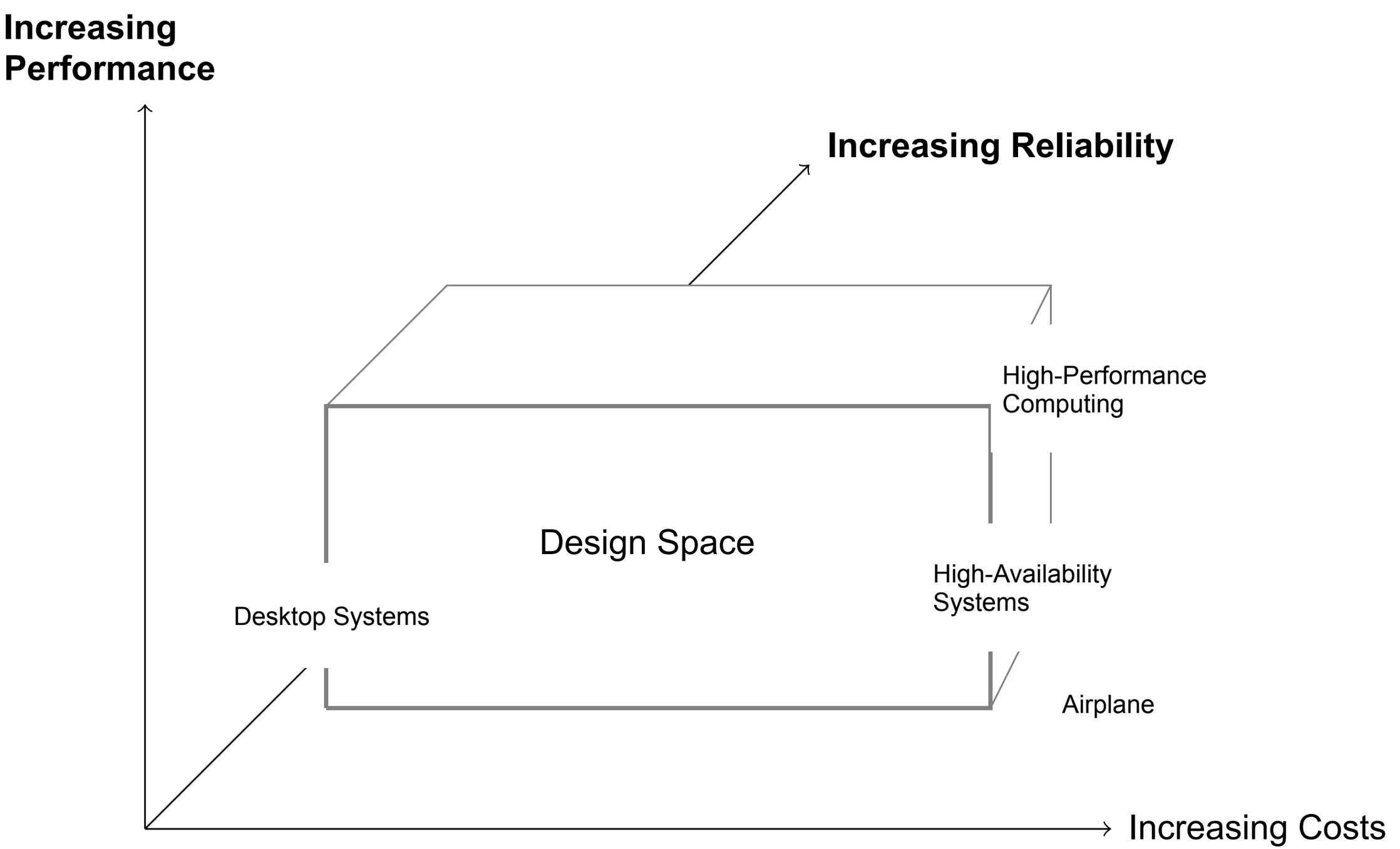
Exascale Reliability

- Let's **just keep** the current **mean time to interrupt (MTTI)**
- 10x - 100x increase in core count =
  - … at least 10x increase in hardware reliability
  - … at least 10x increase in software reliability
  - … at least 10x increase in fault tolerance efficiency



- (Wrong) assumption: No major change in technologies.

Exascale: Two fronts



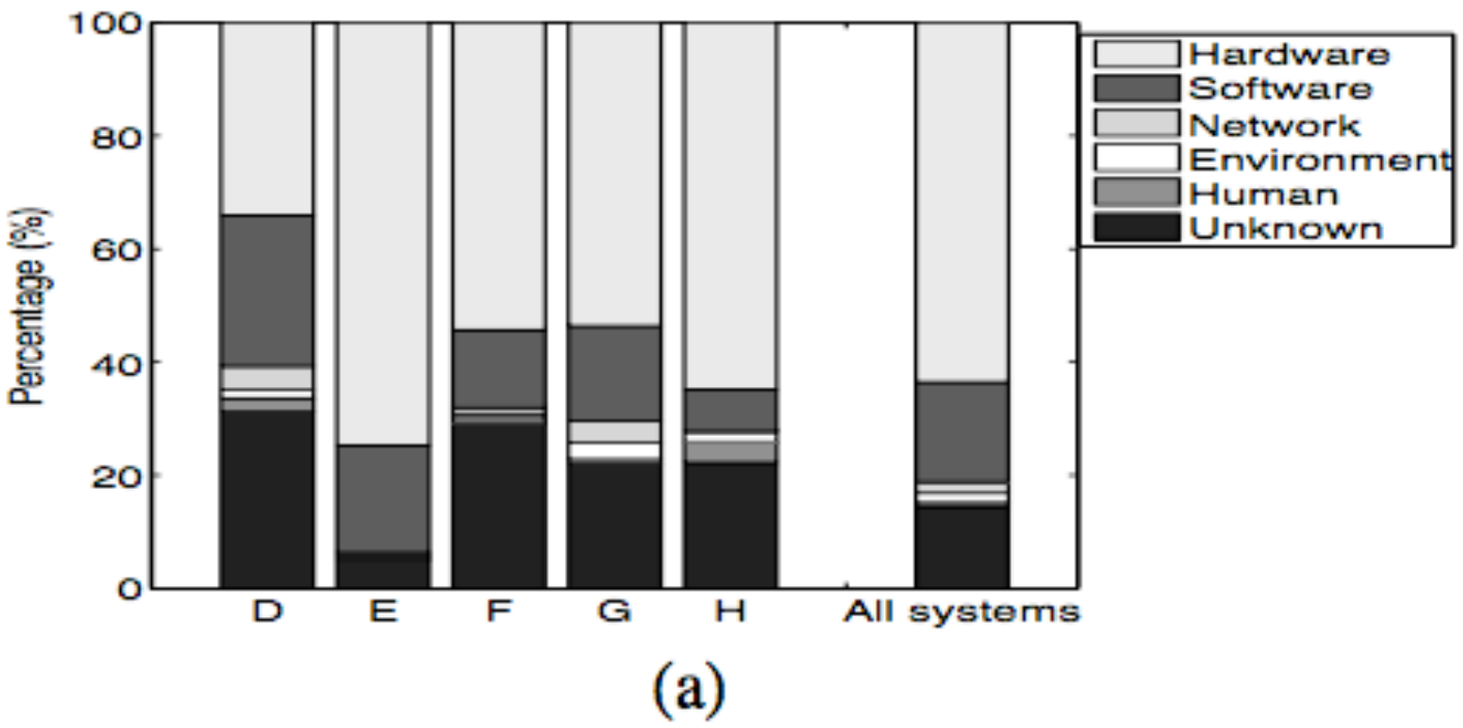
Reliability 101

- **Fault intolerance** - *„They are not acceptable“*
  - Fault prevention - *„Do not introduce them“*
  - Fault removal - *„Remove them“*
- **Fault tolerance** - *„They happen anyway“*
  - Error recovery - *„Heal their effects“*
  - Error mitigation - *„Circumvent their effects“*
- **Models** to understand the **fault -> error -> failure** chain

HPC Reliability - Fault Classes

*„It is important to note that the **number of failures with undetermined root cause is significant**. […] hardware and software are among the largest contributors to failures.“*

[Schröder06]



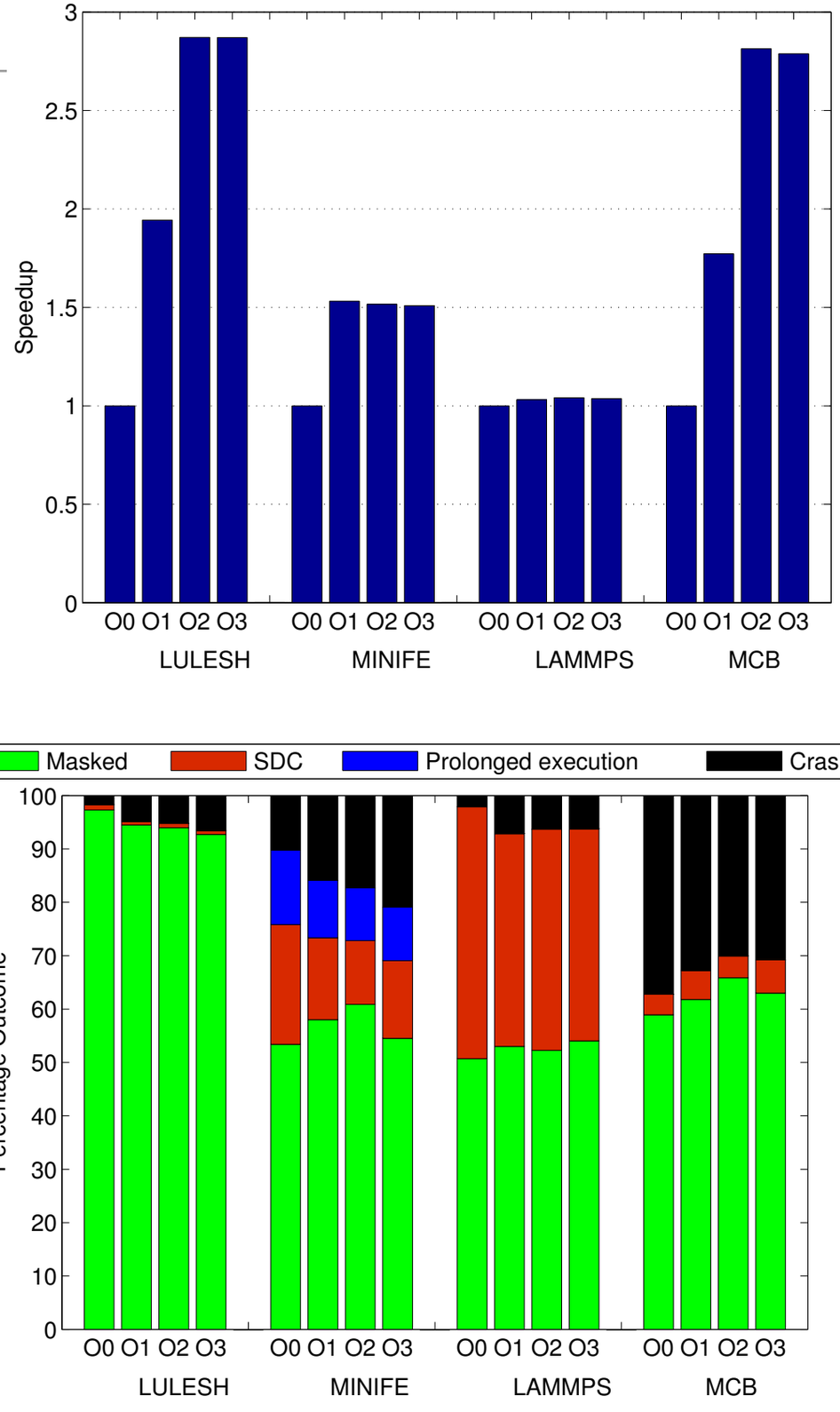
Available data

The table below provides an overview over the availab

Name	Time period	System type
LANL	Dec 96 – Nov 05	HPC clusters
HPC1	Aug 01 – May 06	HPC cluster
HPC2	Jan 04 – Jul 06	HPC cluster
HPC3	Dec 05 – Nov 06	HPC cluster
HPC4	2004 – 2006	HPC cluster
PNNL	Nov 03 – Sep 07	HPC cluster
NERSC	2001 – 2006	HPC cluster
COM1	May 2006	Internet services cluster
COM2	Sep 04 – Apr 06	Internet services cluster
COM3	Jan 05 – Dec 05	Internet services cluster
ask.com	Dec 06 – Feb 07	Internet services cluster
Cray	N/A	Cray systems
Intrepid	Jan 09 – Aug 09	Blue Gene/P

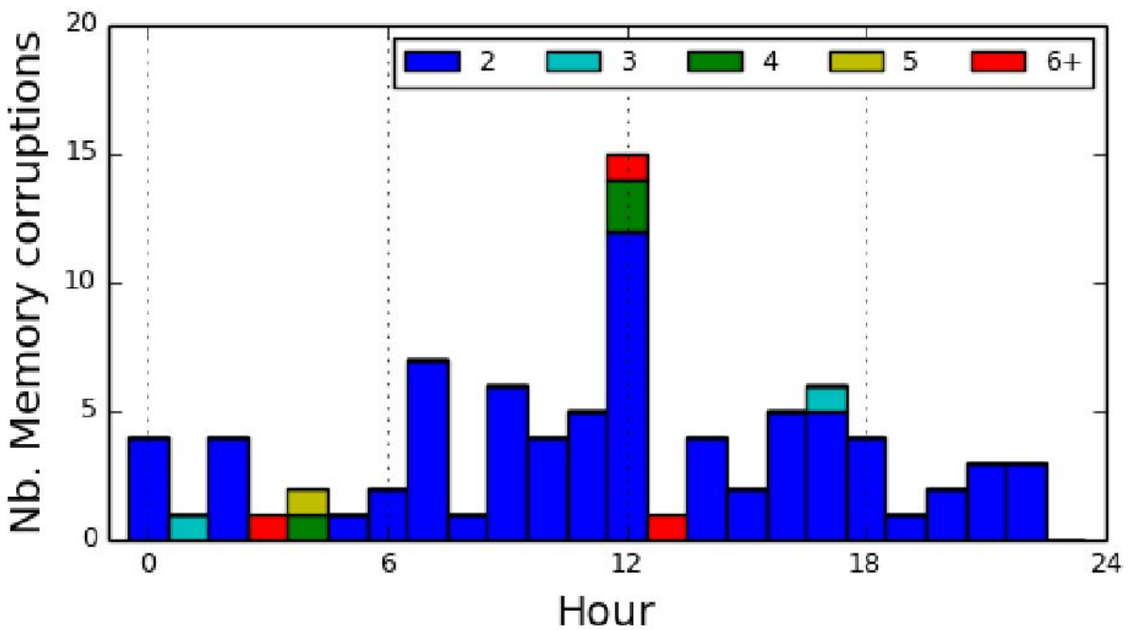
HPC Reliability - Compilers

- Run faster through better code
- Compiler optimizations have a reliability impact [Ashraf17]
  - Simulated register bit flip
  - Compiler-based faul injection
  - Varying increase of failure rates from increasing optimization
  - Blindly applying maximum optimization is no longer feasible



HPC Reliability - DRAM

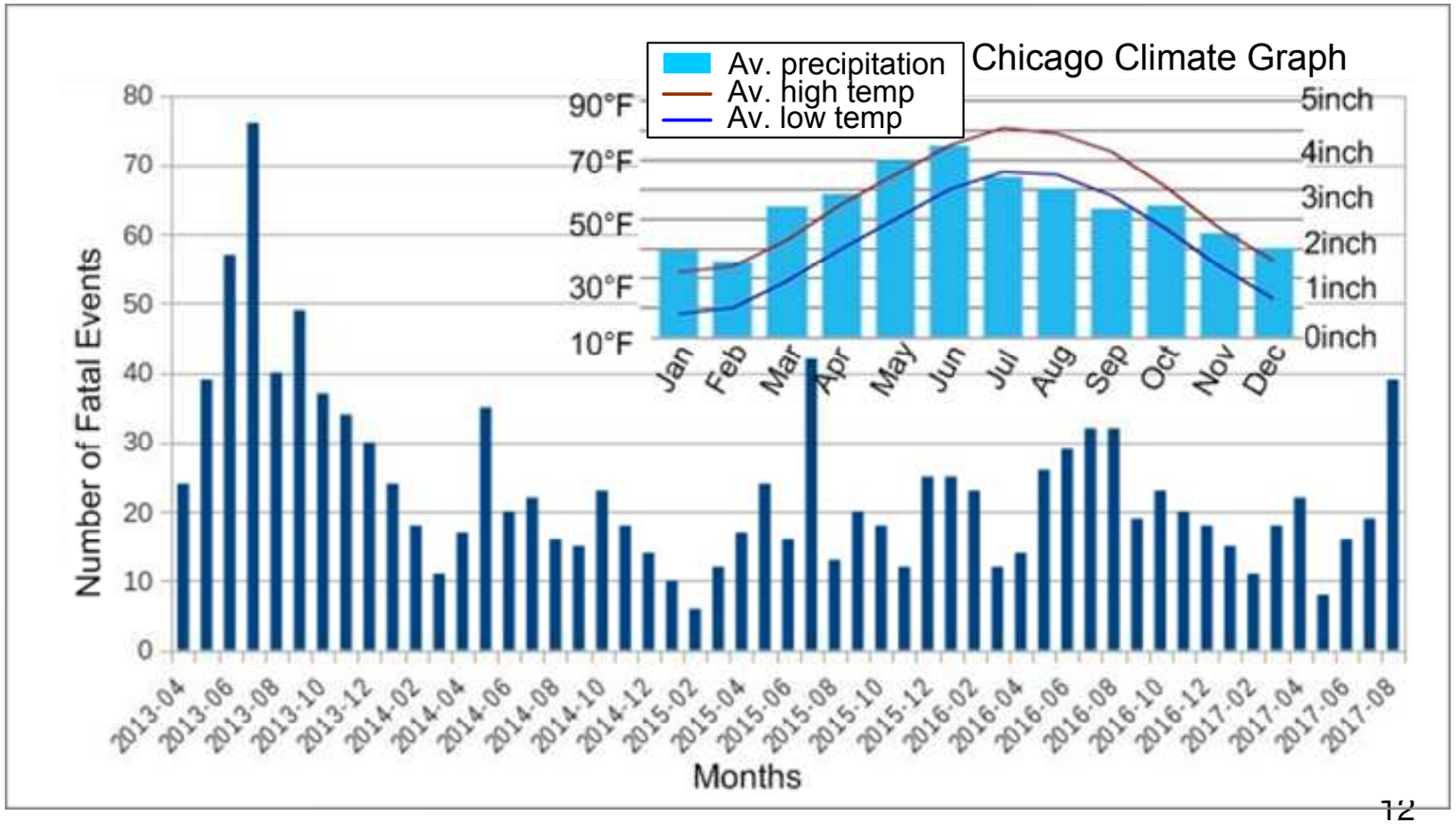
- Soft errors through package pollution, temperature, cosmic rays …
- Example: Analysis with disabled ECC in an HPC cluster [Bautista-Gomez16]
  - 923 nodes, one year, 4.2 million node errors detected, **up to 9 corrupted bits** per word
  - Over 99.9% of errors in less then 1% of the nodes
  - Recommendation: Put nodes immediately into quarantine



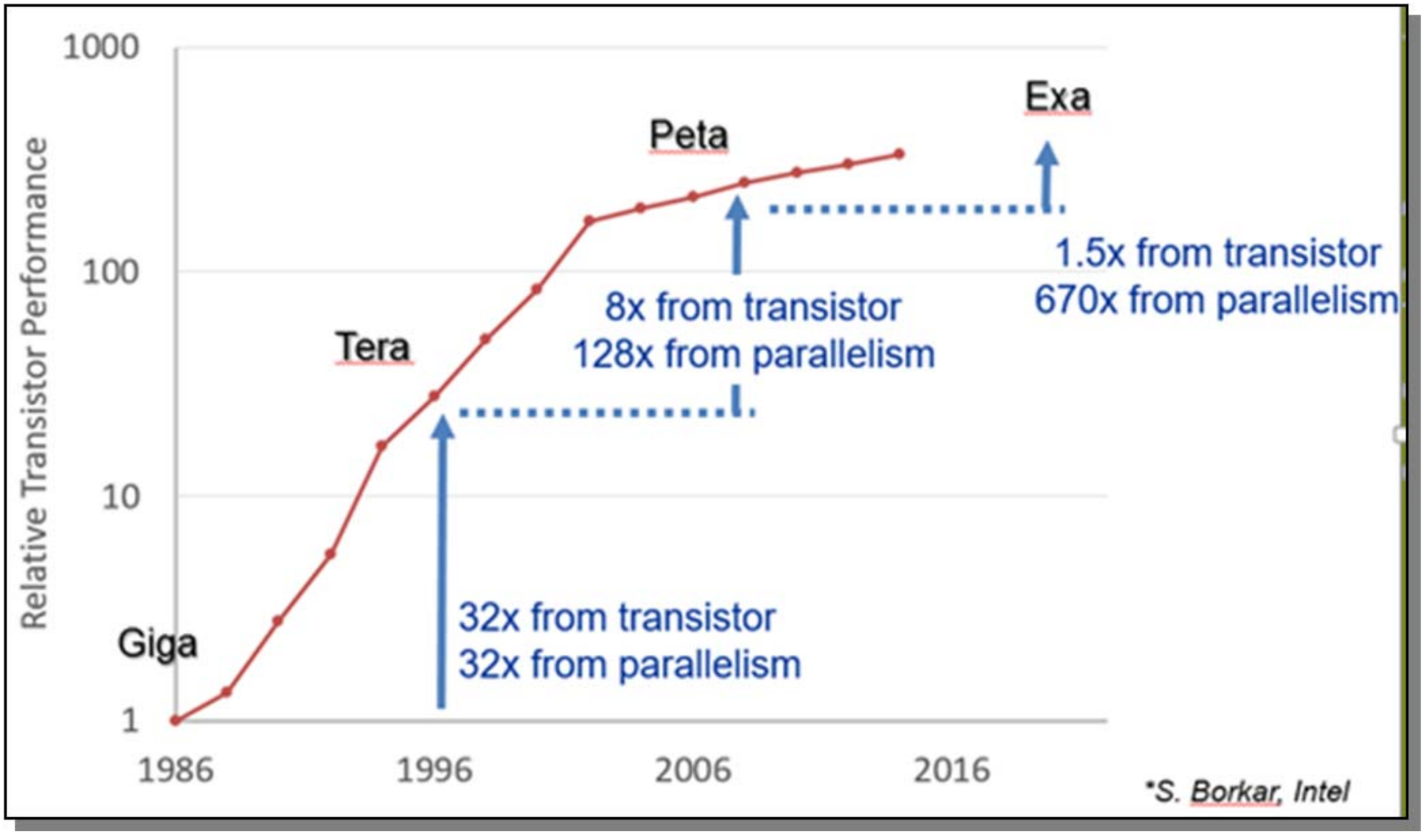
HPC Reliability - Long Term

- 5 years of logs for IBM Blue Gene/Q Mira system [Di18]
- 49.152 nodes, 786.432 cores, PowerPC A2 1.6 GHz
- 80% of fatal events are indicated by ~20% of the monitored attributes

- Strong clustering
- Strong seasonality
- Spatial error correlation mainly inside racks
- MTTI 2-4 days

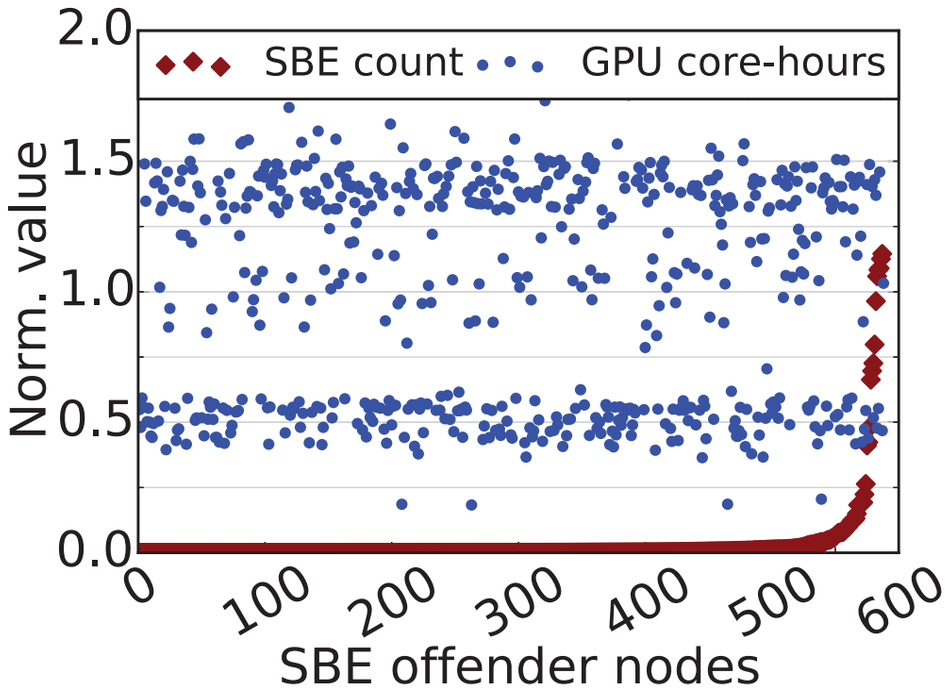


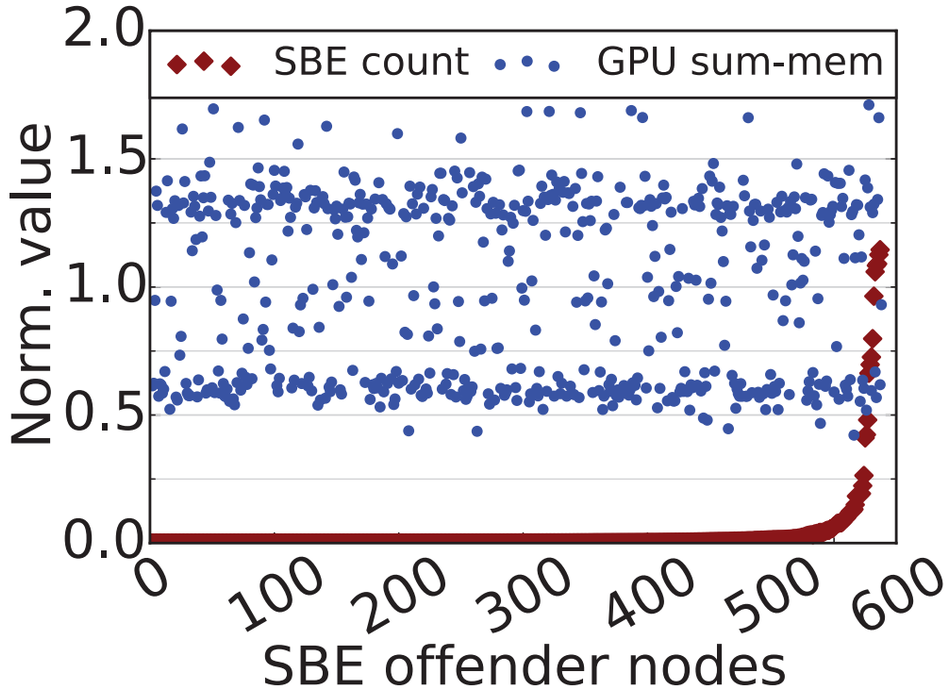
CPU -> GPU


1000
100
10
1
Relative Transistor Performance
Giga
Tera
Peta
Exa
8x from transistor
128x from parallelism
32x from transistor
32x from parallelism
1.5x from transistor
670x from parallelism
1986
1996
2006
2016
*S. Borkar, Intel

HPC Reliability - GPU

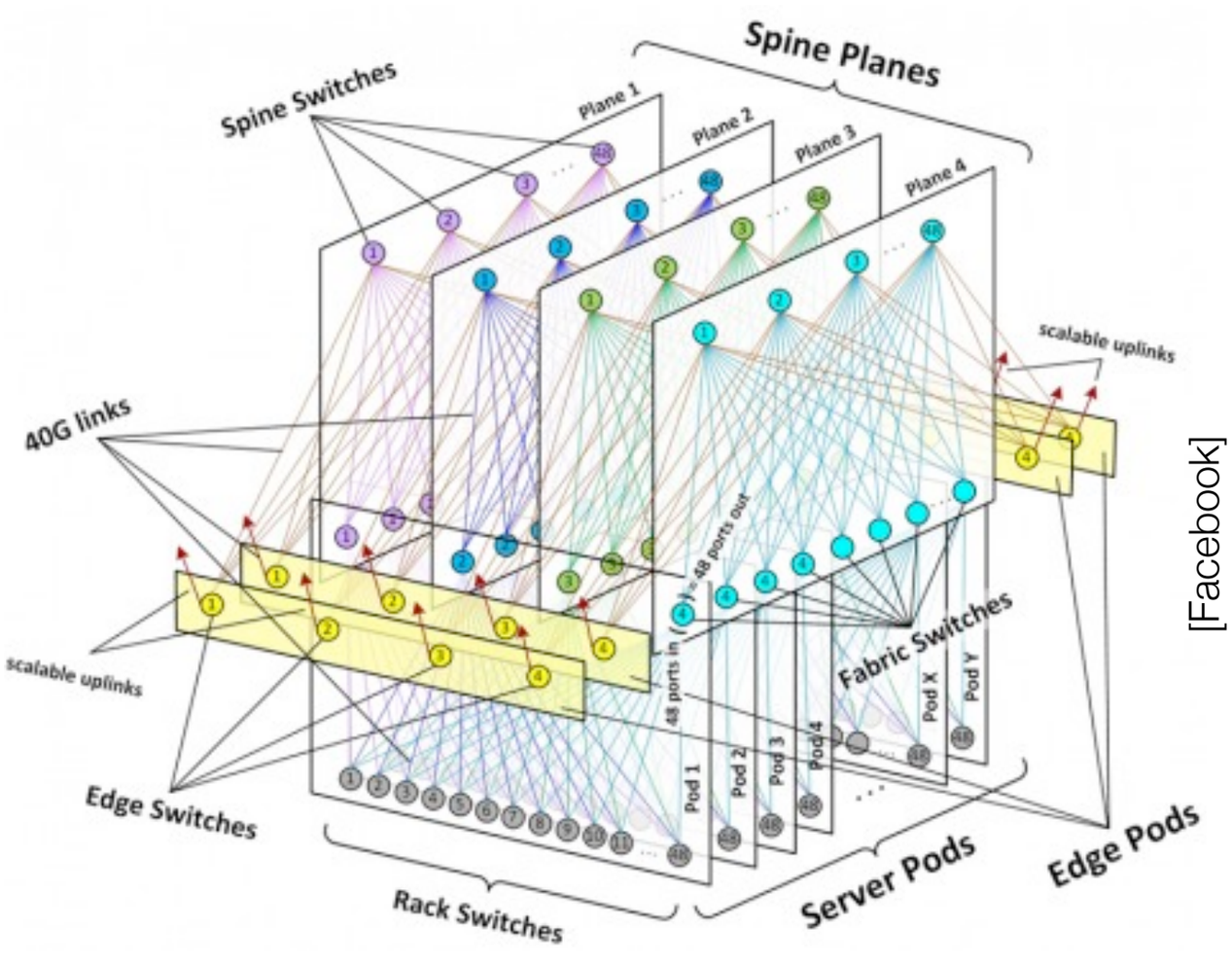
- Titan supercomputer [Nie16]
  - 18.688 x K20X, 6GB memory, 60 Million node hours, 5 months
  - Single-bit-errors do not correlate with core / memory utilization
- Application / user is relevant





Clouds?

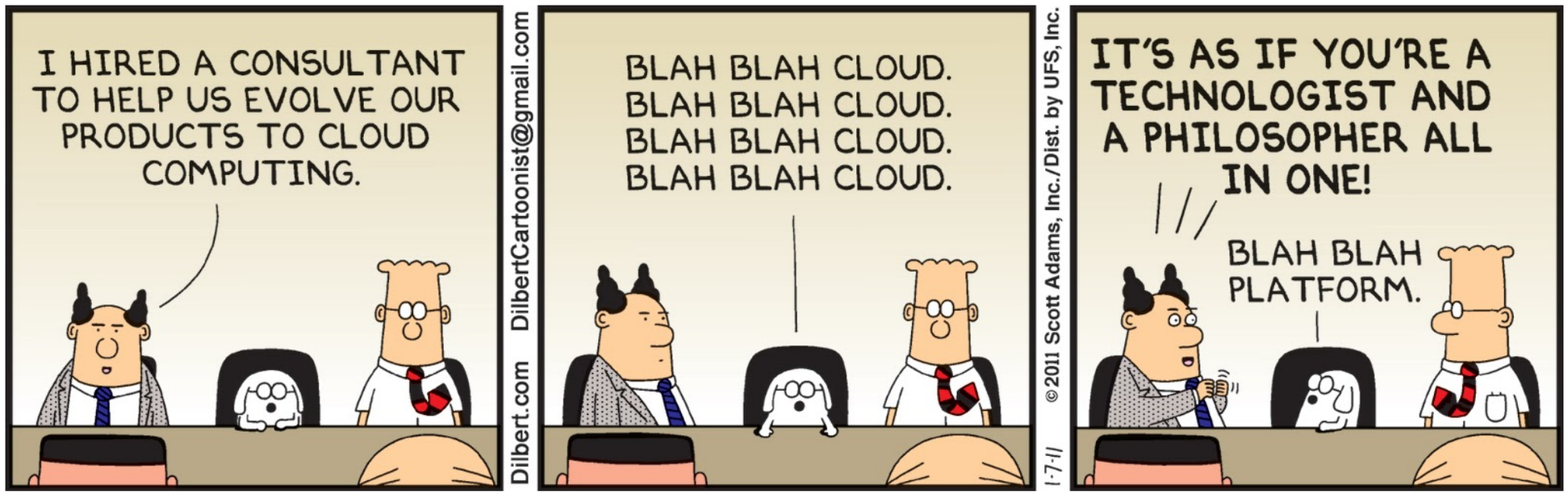
- Approach: Redundancy with cheap hardware
  - Virtualize everything
  - Replicate everything
  - Reduce data consistency
  - Treat errors stochastically
- Load balancing and failover are comparable problems



[Facebook]

Clouds?

- Google & friends solve a different problem
- Cloud: Billions of small single requests, throughput counts
- HPC: Thousands of gigantic single requests, completion counts
- Clouds aim for availability, HPC (still) for reliability



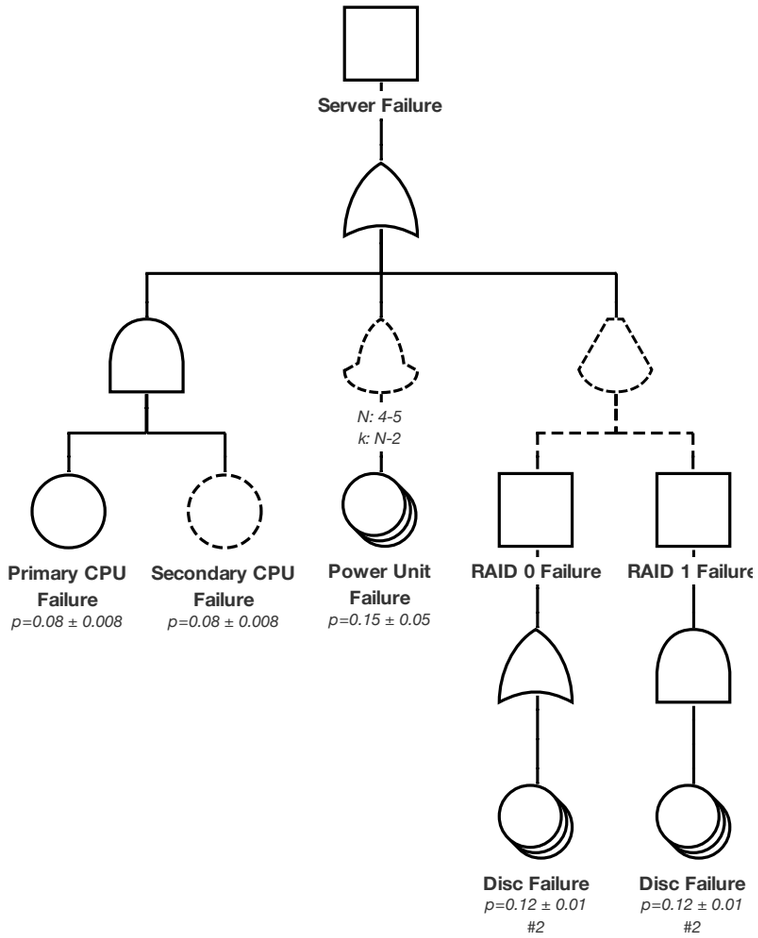
[dilbert.com]

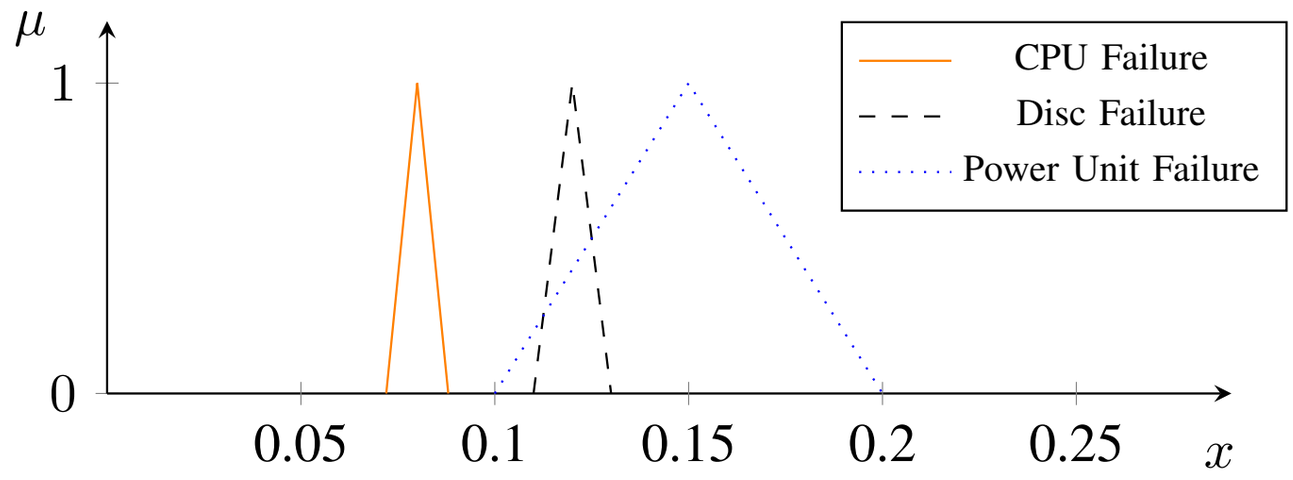
Proposal: Embrace the Uncertainty

- Reliability is getting crucial (again).
- Post-mortem analysis is too late.
- Hardware can no longer solve it alone.
- It does not help to wait.
- Accept that you have no clue about what is going on.
- Create novel ways to deal with this partial system knowledge.
- Make **uncertainty explicit.**

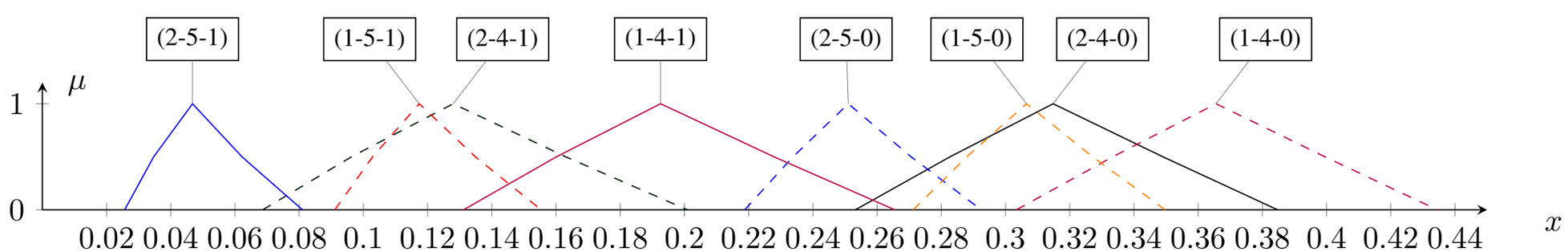
http://amybrucker.com/

Uncertainty: Imprecise Models





FuzzTree	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
(2-4-0)	[.25322, .38475]	[.28271, .34901]	[.31482, .31482]
(1-4-0)	[.30338, .43451]	[.33337, .39946]	[.36558, .36558]
(2-5-0)	[.21875, .29246]	[.2338, .27057]	[.25103, .25103]
(1-5-0)	[.27122, .34969]	[.28792, .3271]	[.30651, .30651]
(2-4-1)	[.06862, .20088]	[.09629, .16302]	[.12796, .12796]
(1-4-1)	[.13118, .26552]	[.16012, .22787]	[.19255, .19255]
(2-5-1)	[.02563, .08101]	[.03467, .06217]	[.04677, .04677]
(1-5-1)	[.09108, .15534]	[.10286, .13484]	[.11738, .11738]



[Tröger13]

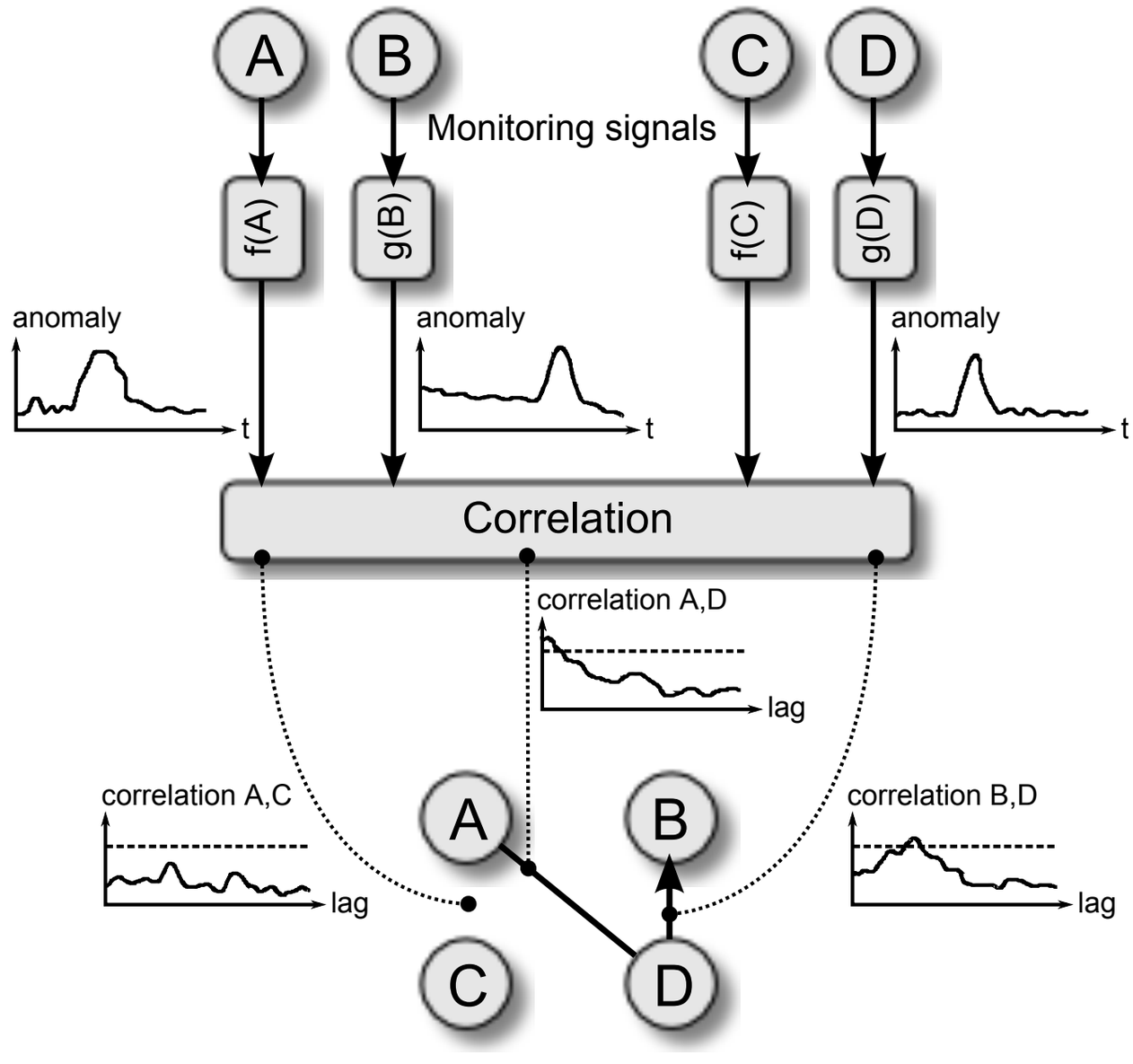
Uncertainty: Resilient Programming

- Processes need to fail-fast
- Applications apply own fault tolerance schemes
  - User level failure mitigation (ULFM)
  - Actor-based message passing
  - Automated instruction redundancy
- Active participation by **HPC users**
- Old codes will most likely **break**

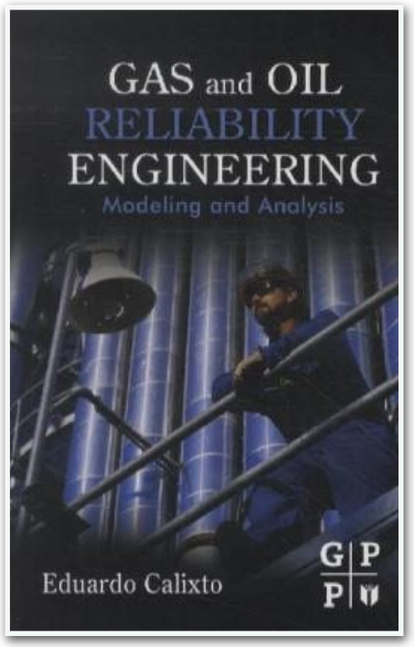
```
save_checkpoint()
result = primary_module()

if acceptance_test(result):
  return result
else:
  load_checkpoint()
  result = alt_module1()

if acceptance_test(result):
  return result
else:
  load_checkpoint()
  result = alt_module2()

if acceptance_test(result):
  return result
else:
  terminate()
```

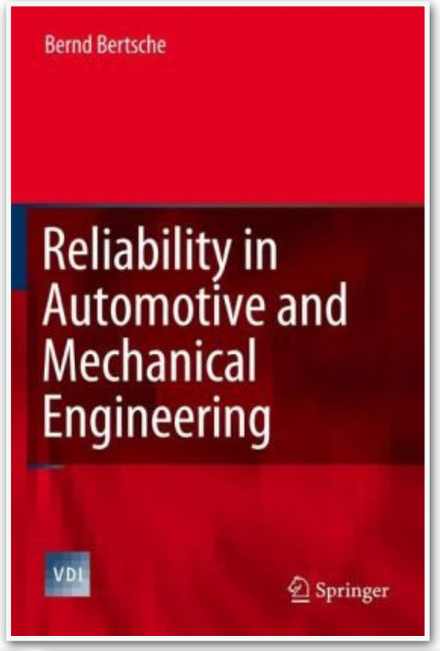
Uncertainty: Anomaly Signals

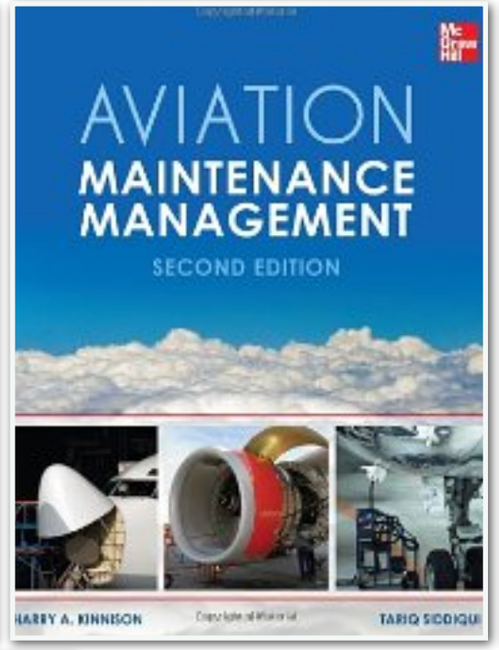
- Anomaly detection approach [Oliner08, Salfner12]
  - Monitoring on different system levels with incompatible metrics
  - Each error situation can best be identified by only one of the system layers
- Idea: Normalize and correlate health indicators across all system levels

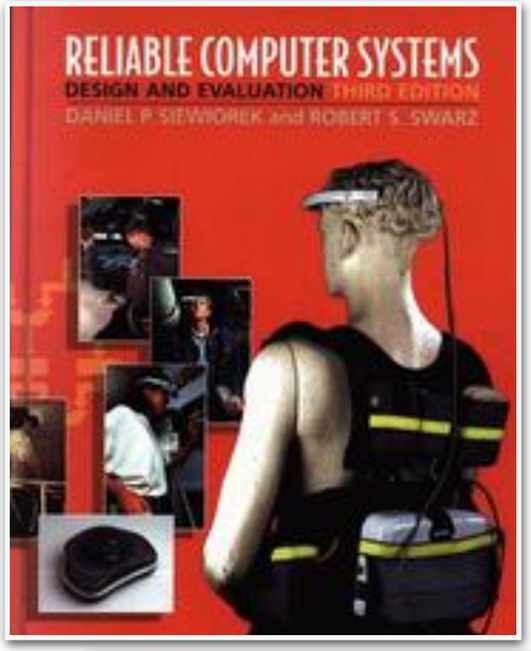


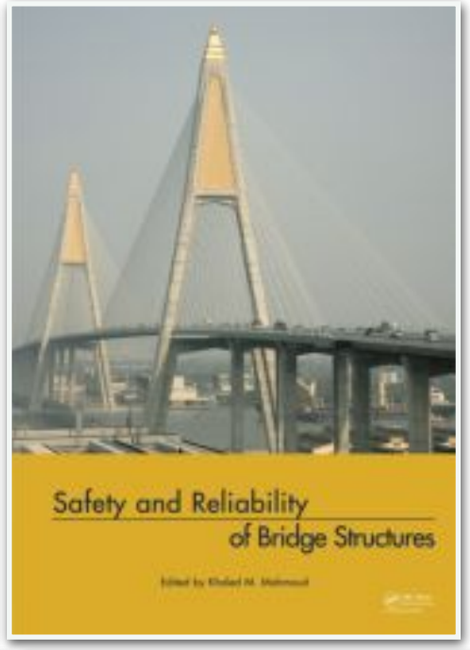
You are not alone …

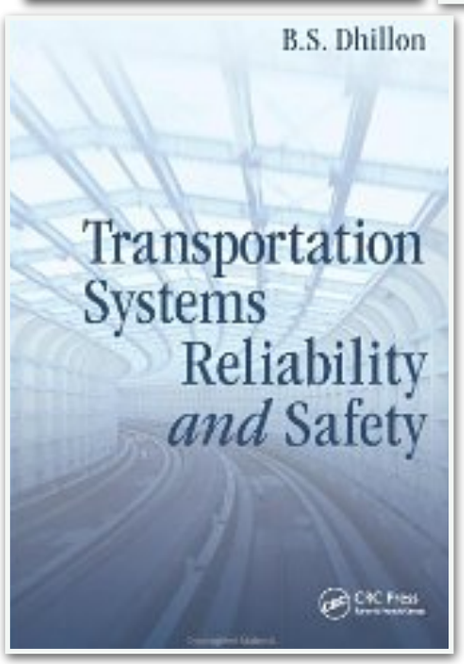


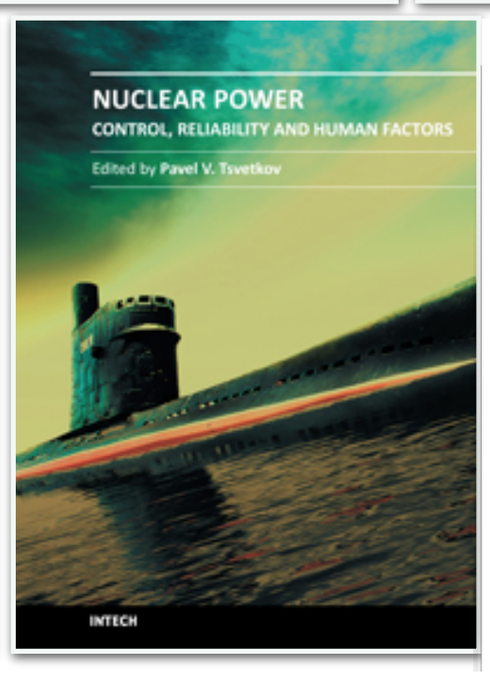


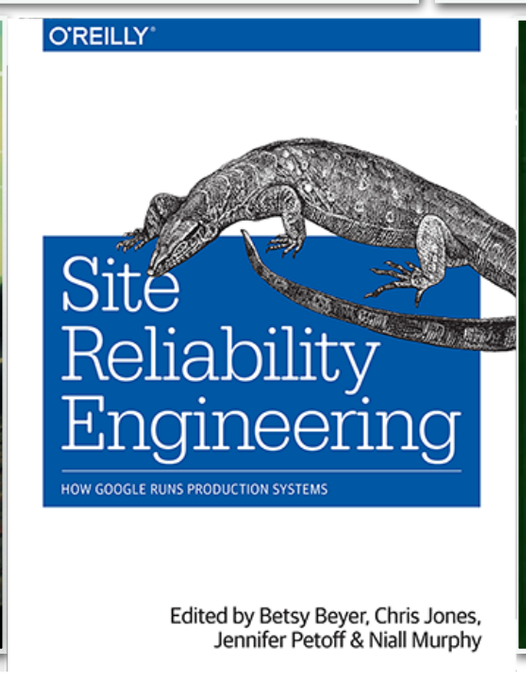


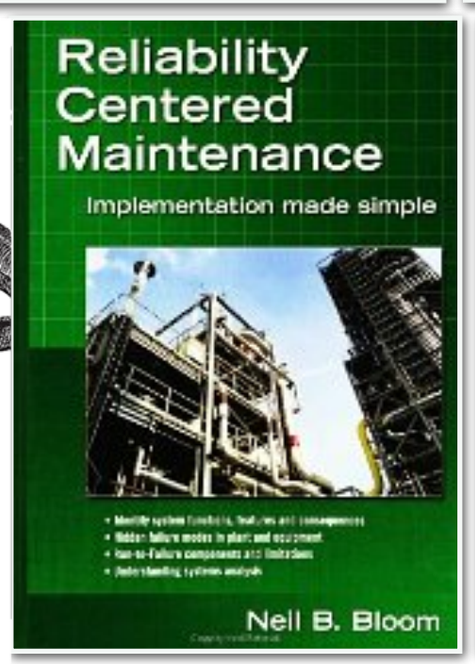


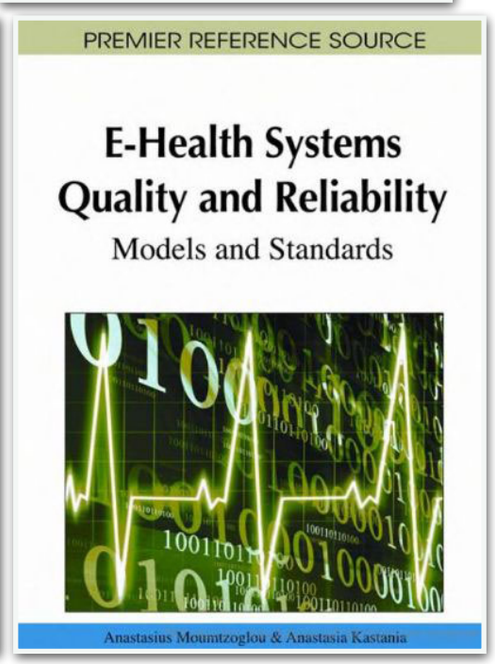






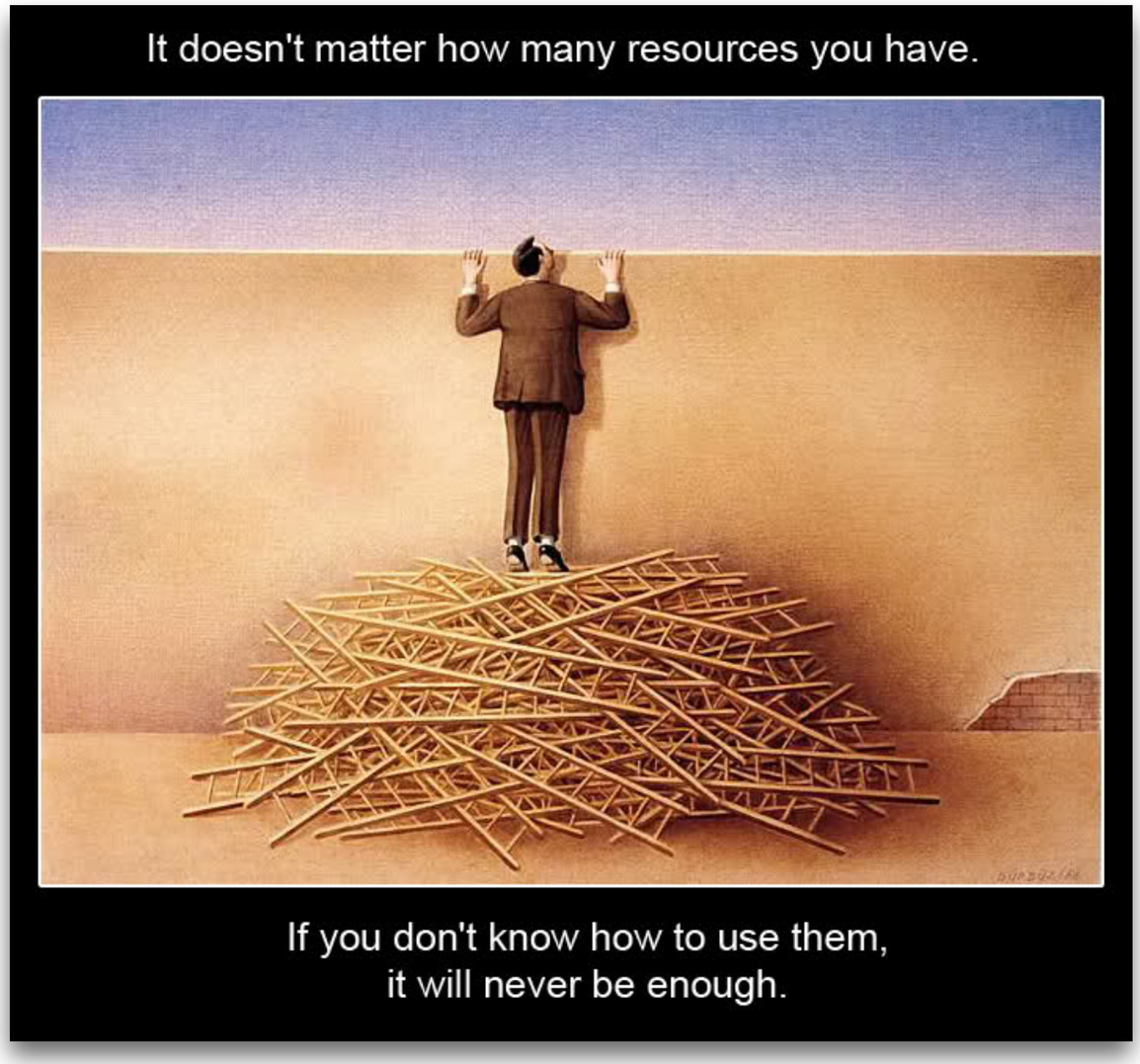






Conclusion

- Reliability must become a first-class citizen (again)
- Proposals:
  - Learn from the non-HPC world
  - Responsibility must move upwards in the stack
  - Make uncertainty explicit, on all layers
  - „Work smarter“



Literature

- Ashraf, R. A., R. Gioiosa, G. Kestor, and R. F. DeMara. 2017. “Exploring the Effect of Compiler Optimizations on the Reliability of HPC Applications.” In 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 1274–83. https://doi.org/10.1109/IPDPSW.2017.7.

- Bautista-Gomez, L., F. Zyulkyarov, O. Unsal, and S. McIntosh-Smith. 2016. “Unprotected Computing: A Large-Scale Study of DRAM Raw Error Rate on a Supercomputer.” In SC ’16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 645–55. https://doi.org/10.1109/SC.2016.54.

- Di, S., H. Guo, R. Gupta, E. R. Pershey, M. Snir, and F. Cappello. 2018. “Exploring Properties and Correlations of Fatal Events in a Large-Scale HPC System.” IEEE Transactions on Parallel & Distributed Systems, 1. https://doi.org/10.1109/TPDS.2018.2864184.

- Nie, B., D. Tiwari, S. Gupta, E. Smirni, and J. H. Rogers. 2016. “A Large-Scale Study of Soft-Errors on GPUs in the Field.” In 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), 519–30. https://doi.org/10.1109/HPCA.2016.7446091.

- Oliner, A. J., A. Aiken, and J. Stearley. 2008. “Alert Detection in System Logs.” In , 959–64. https://doi.org/10.1109/ICDM.2008.132.

- Pickartz, S., S. Lankes, A. Monti, C. Clauss, and J. Breitbart. 2016. “Application Migration in HPC — A Driver of the Exascale Era?” In 2016 International Conference on High Performance Computing Simulation (HPCS), 318–25. https://doi.org/10.1109/HPCSim.2016.7568352.

- Römer, Paul, and Peter Tröger. 2011. “Reliability Implications of Register Utilization: An Empirical Study.” In IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC). https://doi.org/10.1109/DASC.2011.41.

- Salfner, Felix, and Peter Tröger. 2012. “Predicting Cloud Failures Based on Anomaly Signal Spreading.” In 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Boston.

- Schroeder, Bianca, and Garth A. Gibson. 2006. “A Large-Scale Study of Failures in High-Performance Computing Systems.” In , 249–58. Washington, DC, USA. http://dx.doi.org/10.1109/DSN.2006.5.

- Tröger, Peter. 2018. Unsicherheit und Uneindeutigkeit in Verlässlichkeitsmodellen. Springer Vieweg. //www.springer.com/de/book/9783658233402.

- Tröger, Peter, Franz Becker, and Felix Salfner. 2013. “FuzzTrees - Failure Analysis with Uncertainties.” In 19th Pacific Rim International Symposium on Dependable Computing, 263–72. IEEE. https://doi.org/10.1109/PRDC.2013.48.