Designing Smart and Resilient Extreme-Scale Systems

Christian Engelmann, Ph.D.

Senior Scientist & Group Leader
Intelligent Systems and Facilities Group
Advanced Computing Systems Research Section
Computer Science and Mathematics Division
Oak Ridge National Laboratory

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Recent Anecdotes

- 11,000 out of 18,800 GPUs had to be replaced in ORNL's Titan supercomputer in 2016-18 due to a serious reliability issue
- Google in 2021 reported that certain processor cores compute wrong results, such that some encrypted data could only be decrypted on cores it was encrypted on
- Meta (Facebook Engineering) revealed in 2021 experiences with silent data corruption at scale
- A 2022 New York Times article, titled "*Chip Errors Are Becoming More Common and Harder to Track Down*", detailed some of the ongoing issues with computer hardware reliability

The HPC Resilience Challenge

- Resilience in extreme-scale supercomputers is an optimization problem between the key design and deployment cost factors:
  - Performance, resilience, and power consumption
- The challenge is to build a reliable system within a given cost budget that achieves the expected performance
- This requires fully understanding the resilience problem and offering efficient resilience mitigation technologies
  - What is the fault model of such systems?
  - What are realistic expectations for reliability?
  - What is the impact of faults on applications?
  - How can mitigation in hard-/software help and at what cost?

Characterizing Supercomputer Faults, Errors and Failures

Novel Ideas:

- Applies a unified taxonomy for supercomputer faults, errors and failures
- Understanding resilience is a data analytics problem, requiring fusion and analysis of different logs and system health data

Impact:

- Develops an understanding of observed and inferred supercomputer reliability conditions
- Extrapolates this knowledge to future systems
- Enables the systematic improvement of resilience in extreme-scale systems
- Keeps applications running to a correct solution in a timely and efficient manner in spite of frequent faults, errors, and failures

Accomplishments:

- Analyzed 1.2 billion node hours of logs from the Jaguar, Titan, and Eos systems at OLCF
- Developed tools for analyzing logs and creating a fault, error and failure catalog
- Created novel modeling techniques to characterize temporal and spatial failure behavior

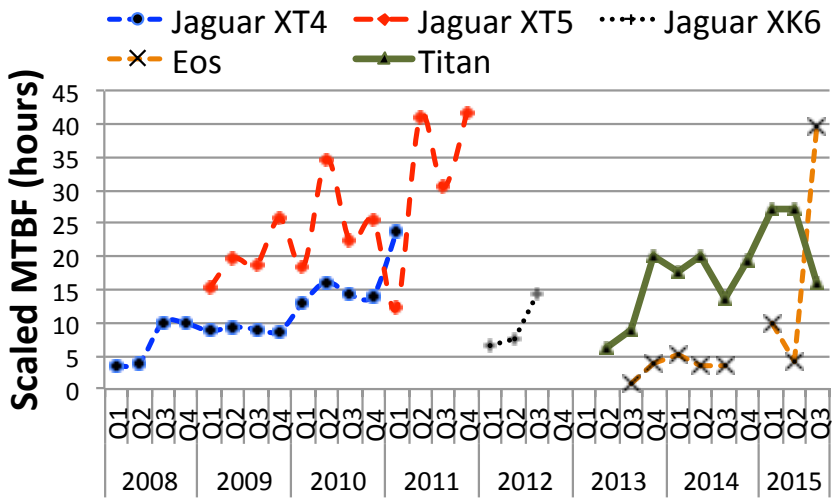
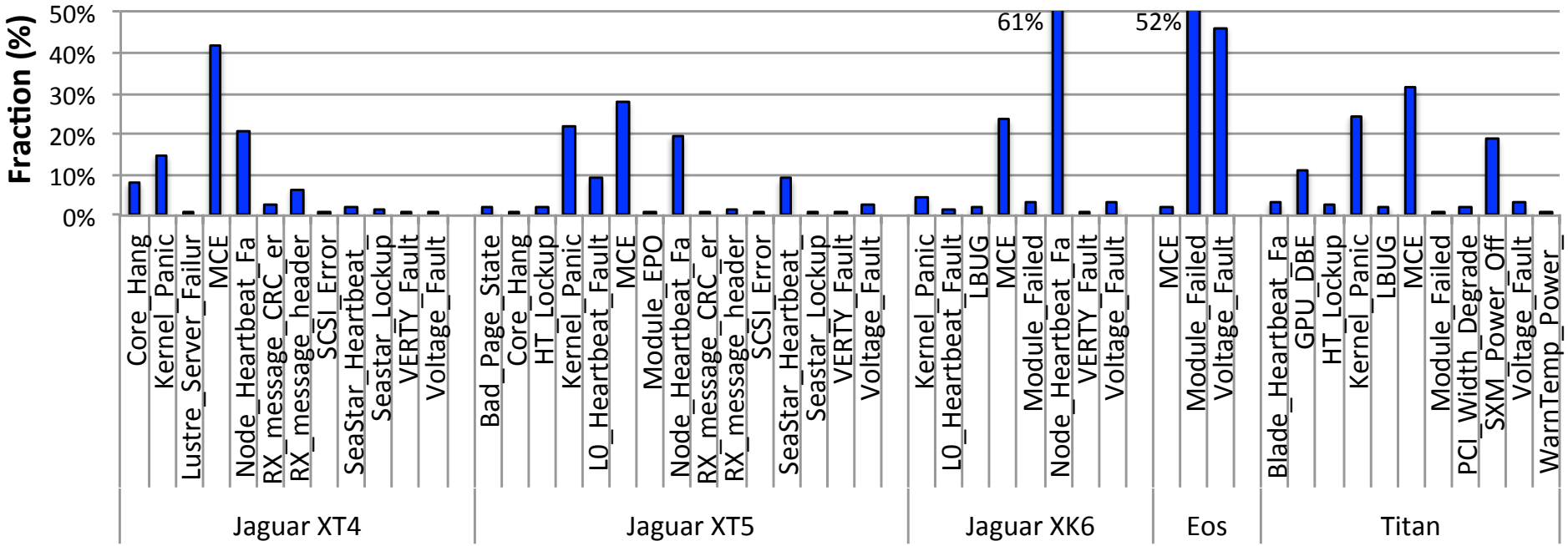


Figure: Each system goes through phases of high and low stability due to continuous efforts of system administrators to improve overall system reliability

Oak Ridge National Laboratory

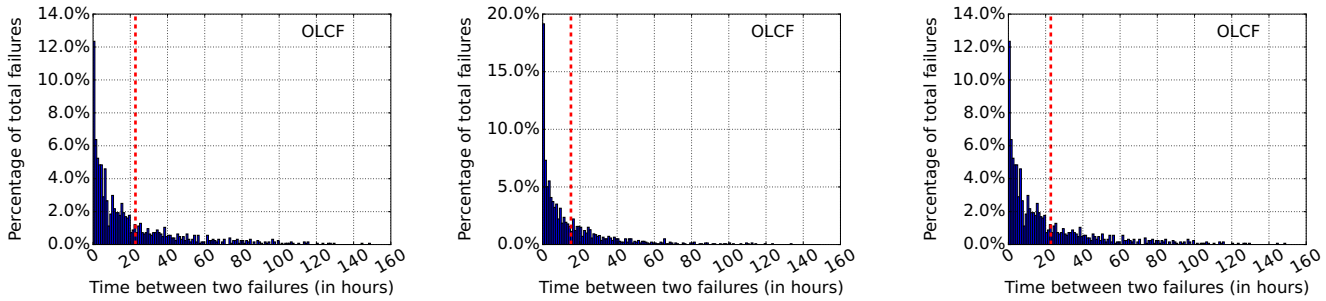
Saurabh Gupta, Devesh Tiwari, Tirthak Patel, and Christian Engelmann. **Reliability of HPC systems: Large-term Measurement, Analysis, and Implications.** SC'17. DOI 10.1145/3126908.3126937.

Characterizing Supercomputer Faults, Errors and Failures



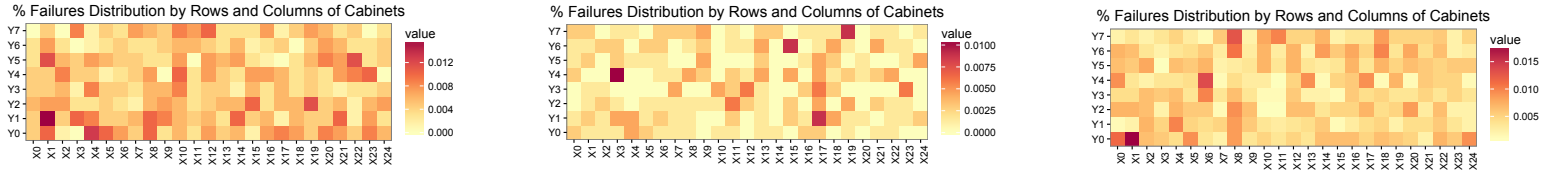
Fraction of each failure type on the studied systems

Characterizing Supercomputer Faults, Errors and Failures



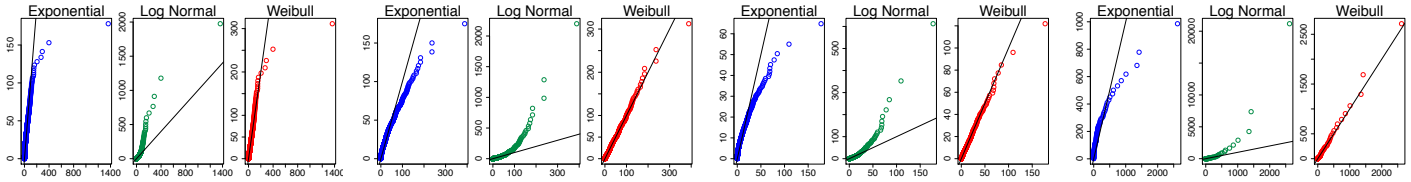
(a) Jaguar XT4 (b) Titan (c) Eos

Failure inter-arrival time for 3 studied systems (MTBF as red vertical line)



(a) Jaguar XT5 (b) Jaguar XK6 (c) Titan

Spatial distribution of failures among cabinets for 3 studied systems

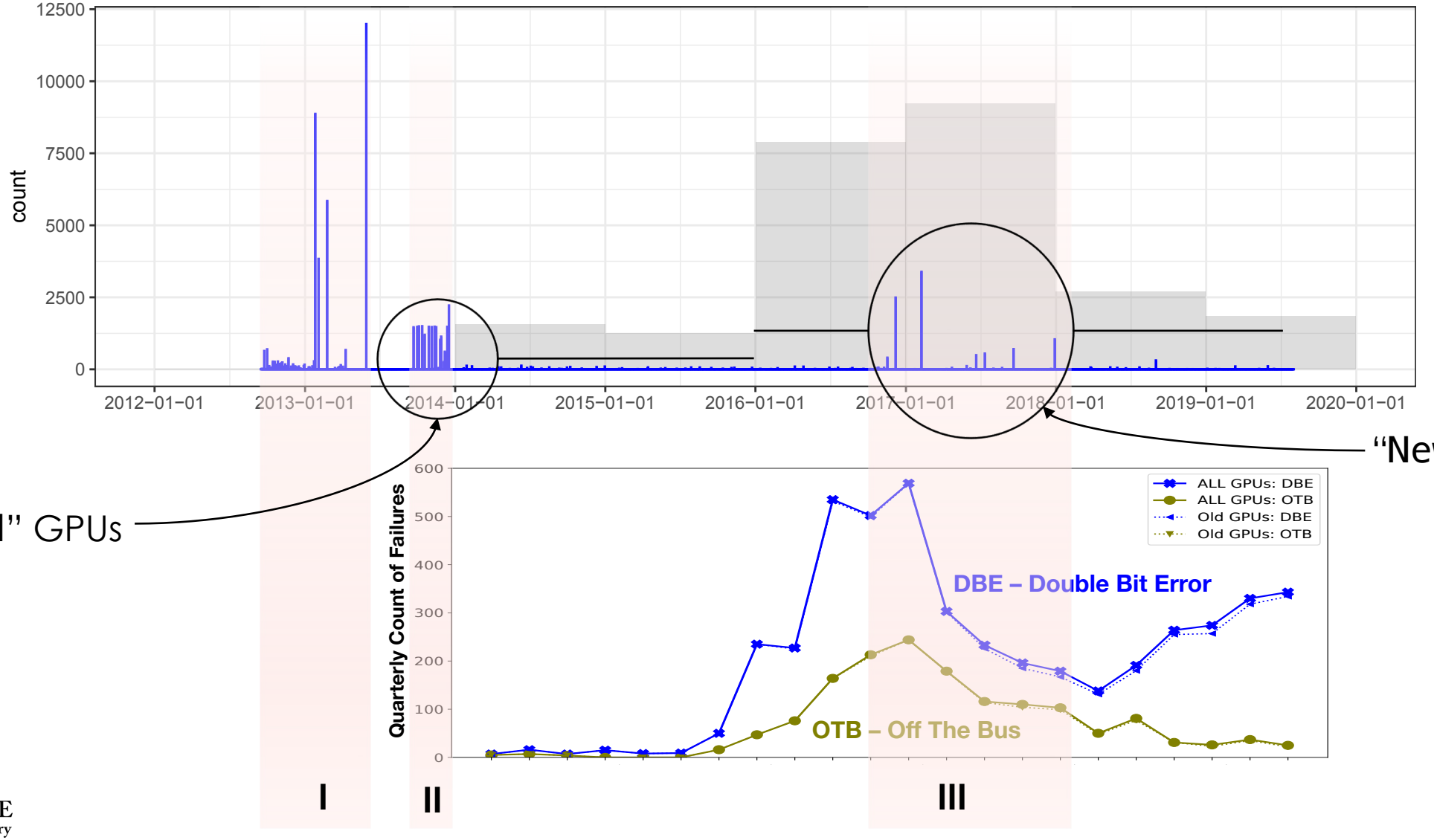


(a) Jaguar XT4 (b) Jaguar XT5 (c) Jaguar XK6 (d) Eos

QQ-plots showing goodness of fit for the failure inter-arrival times for 4 studied systems with different failure probability density functions (Weibull fits best)

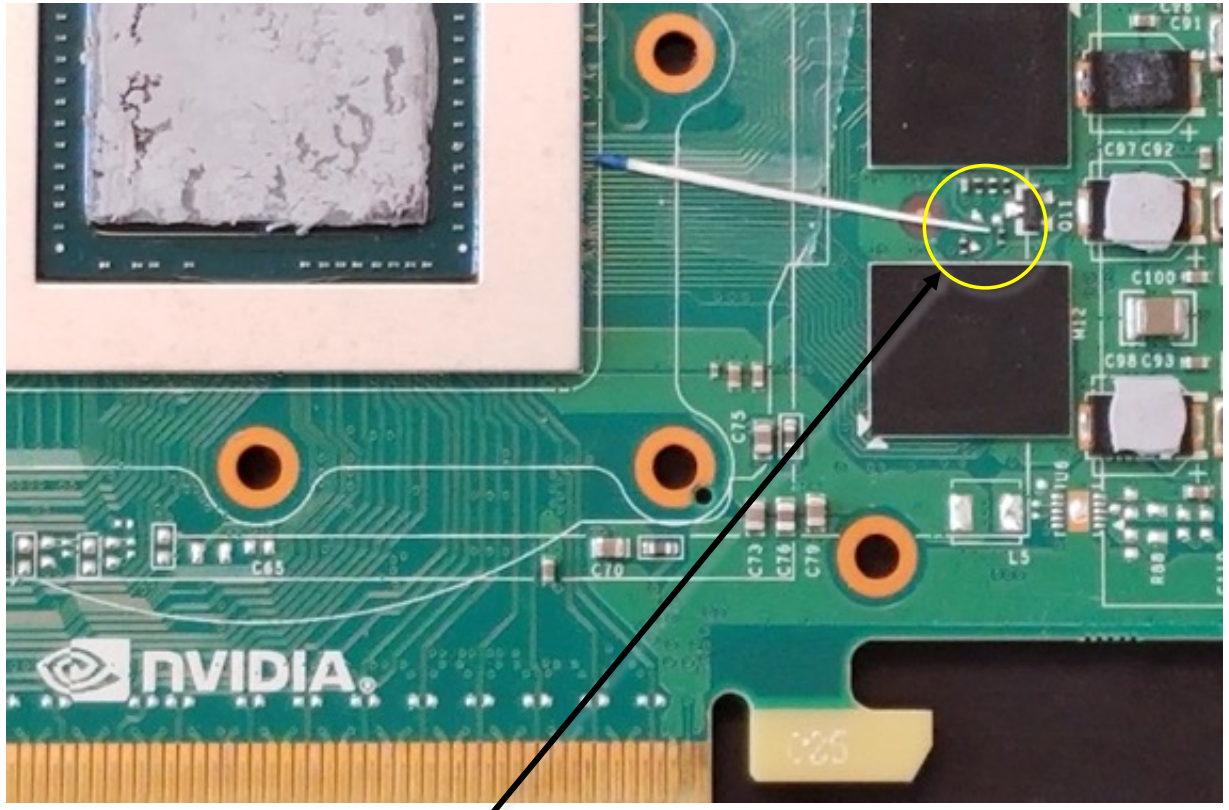
GPU Failures and Replacements in ORNL's Titan

GPU swaps detected at inventories (narrow blue) and yearly sum totals for 2014 and later (wide gray)

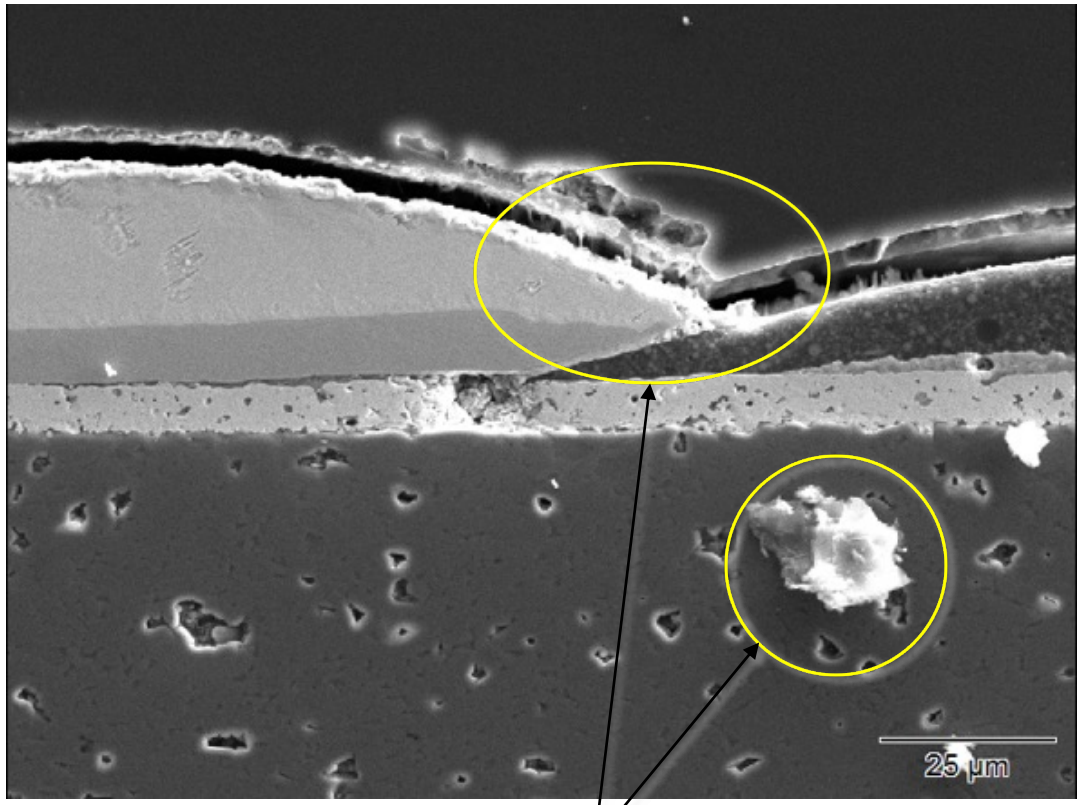


OAK RIDGE National Laboratory

Root Cause: Non-ASR Components on SXM GPU



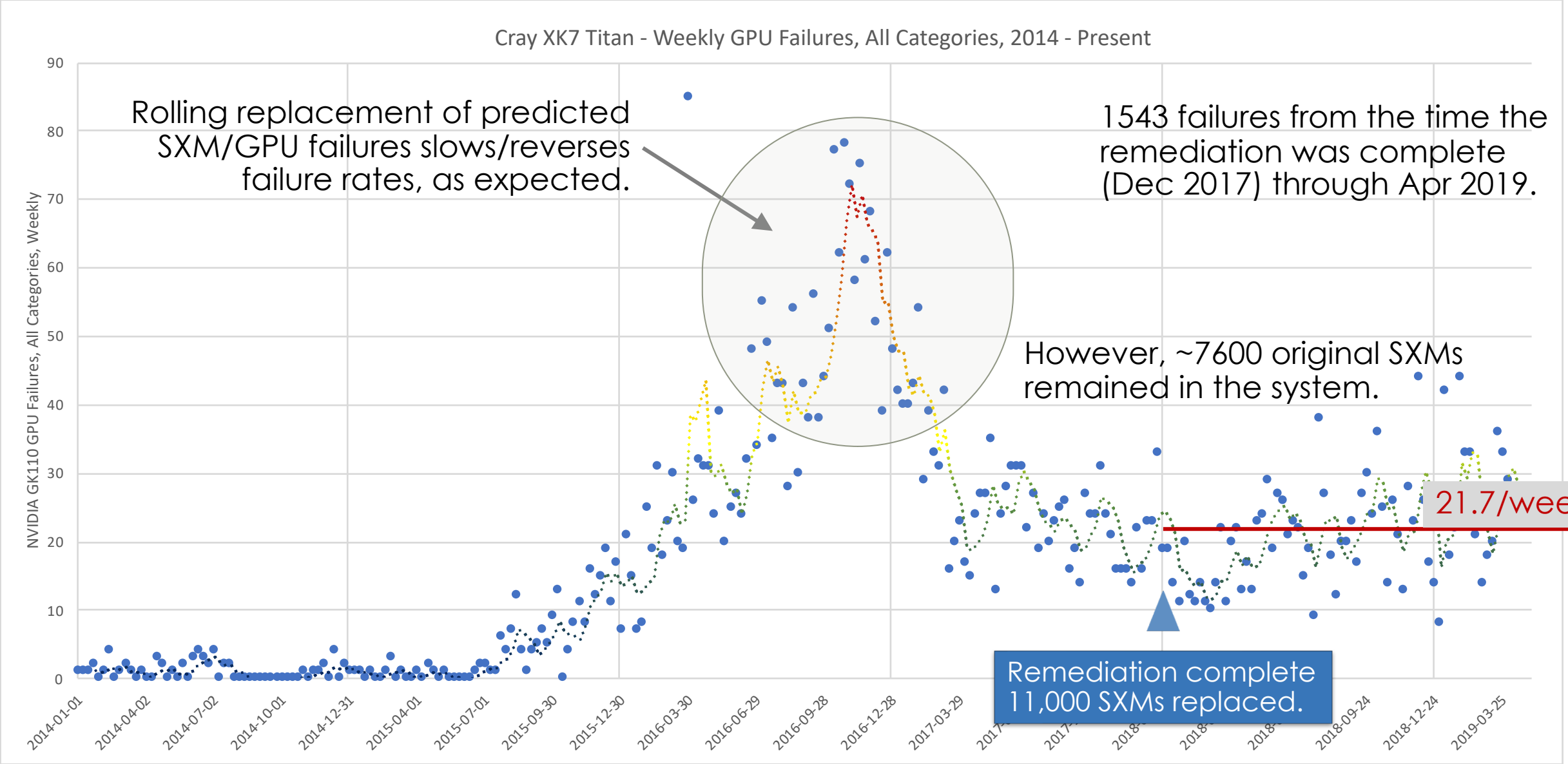
NVIDIA SXM – Location of a non-ASR



Silver-sulfide corrosion
"Flowers-of-Sulfur"

ASR = Anti-Sulfur Resistor

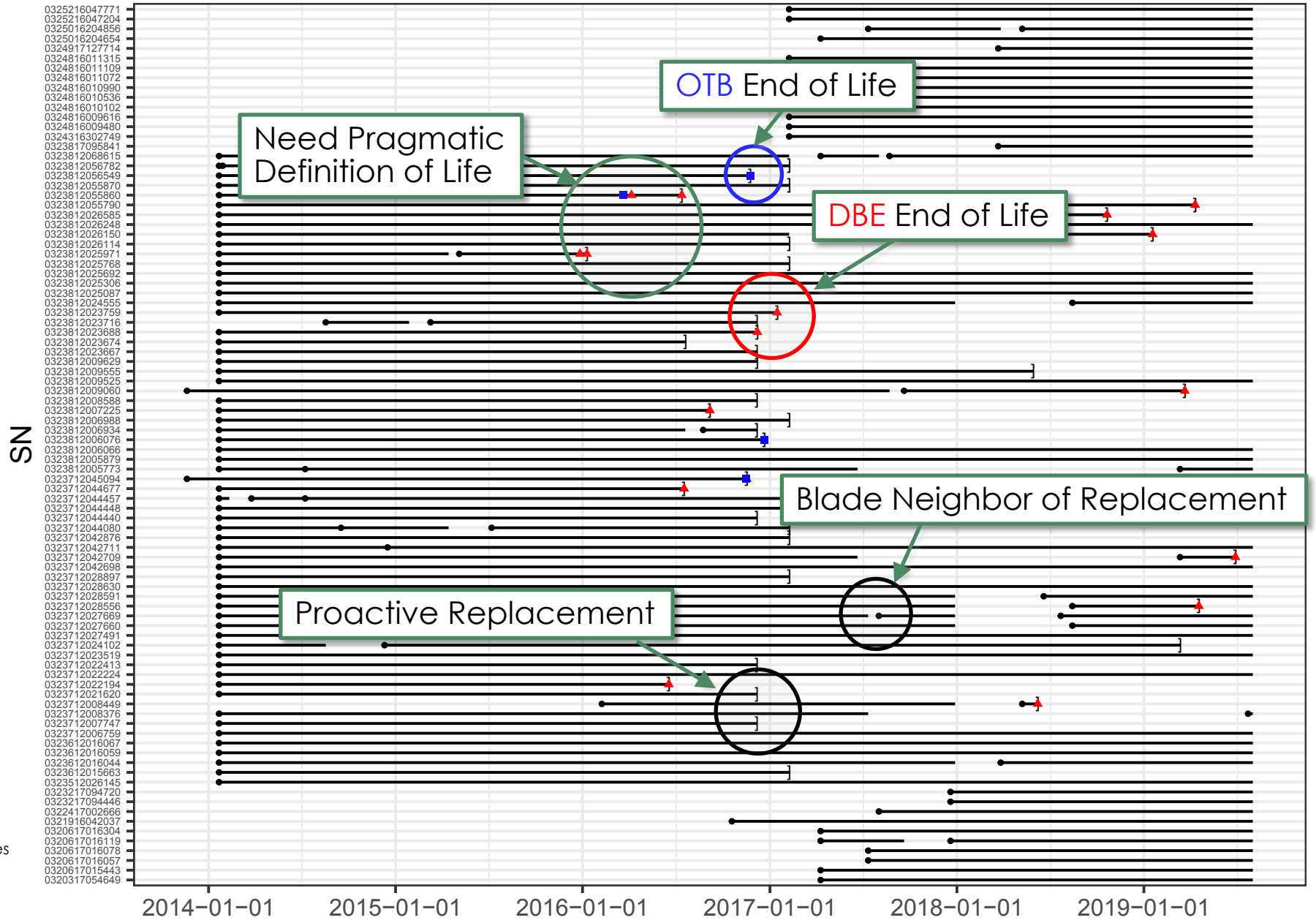
Cray XK7 Titan – Weekly GPU Failures



GPU Life Visualization: Serial Number View

Critical for:

- Understanding data
- Defining GPU Life
- Data processing verification

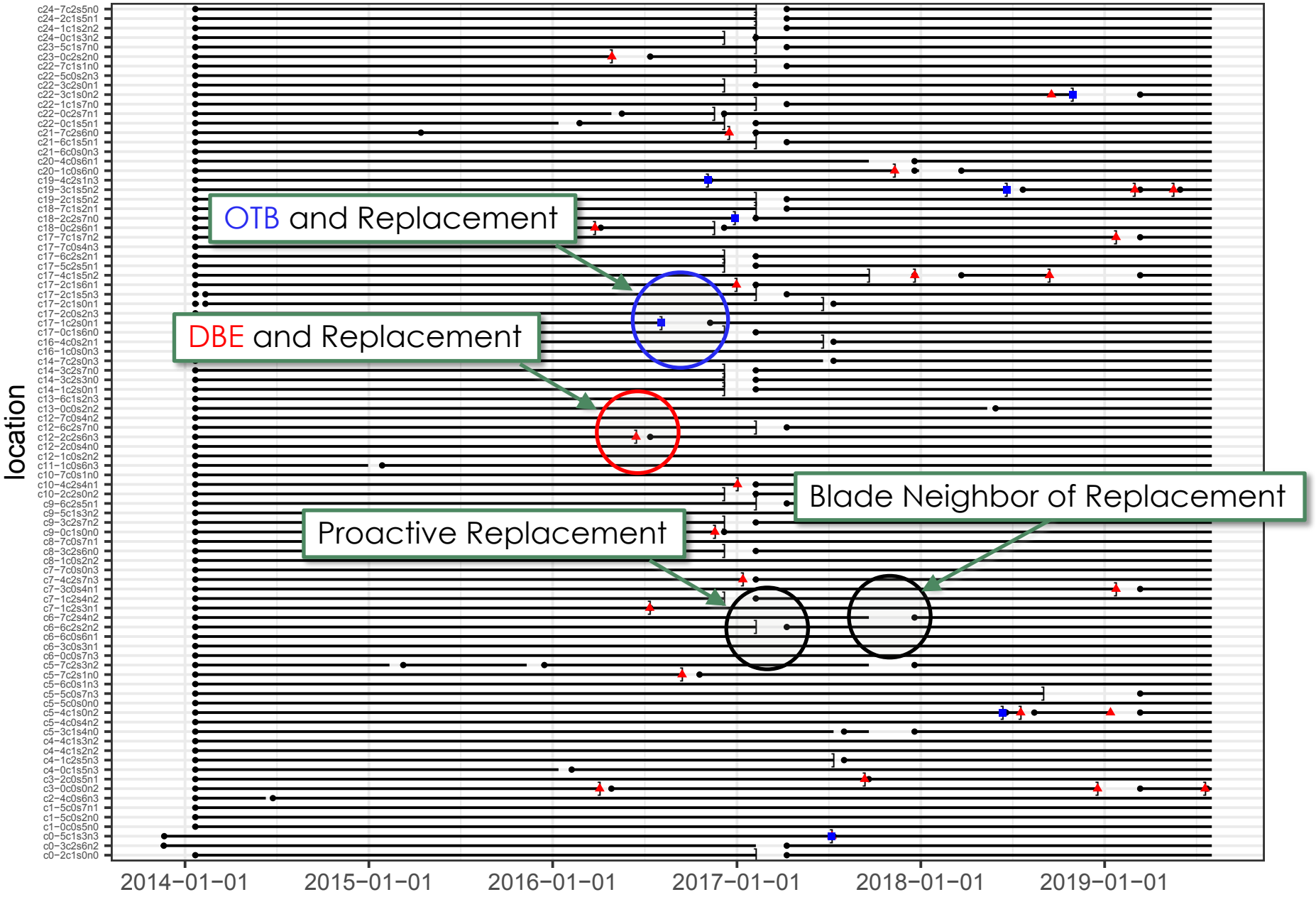


Produced in R via `ggplot2` and `lubridate` packages

GPU Life Visualization: Location View

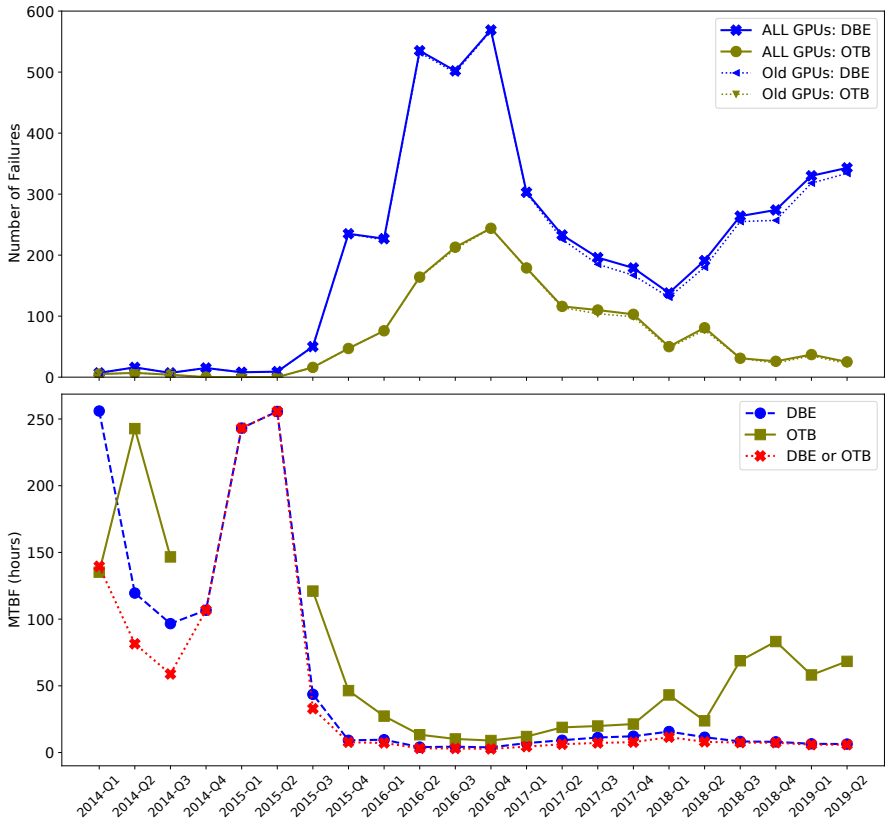
Critical for:

- Understanding data
- Defining GPU Life
- Data processing verification

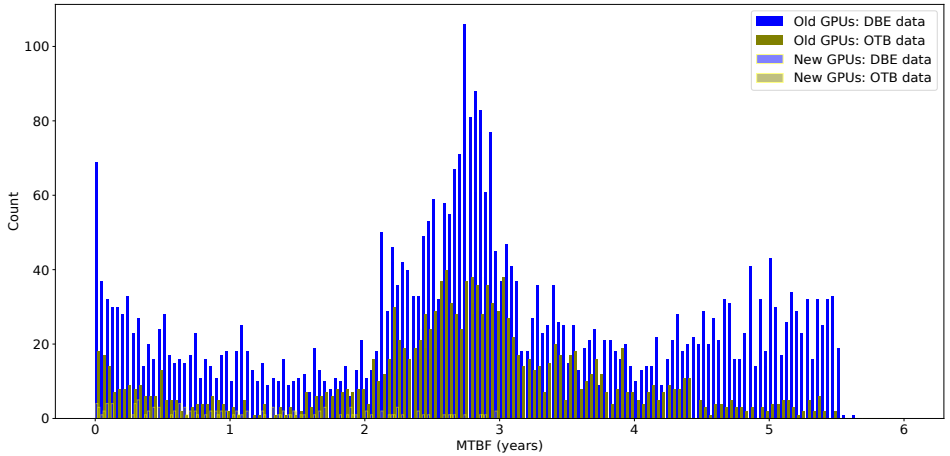


Produced in R via `ggplot2` and `lubridate` packages

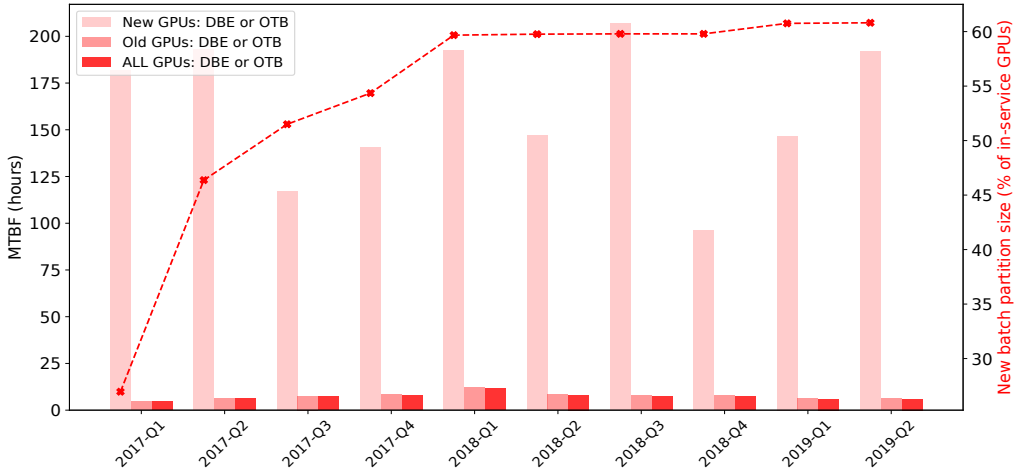
Traditional Reliability in HPC is Focused on MTBF



System-wide Reliability: Quarterly number of failures (top) and MTBF (bottom).



Individual GPU Reliability: MTBF histogram for units that had at least one failure. Interpret carefully: lacks information from units with no failures!

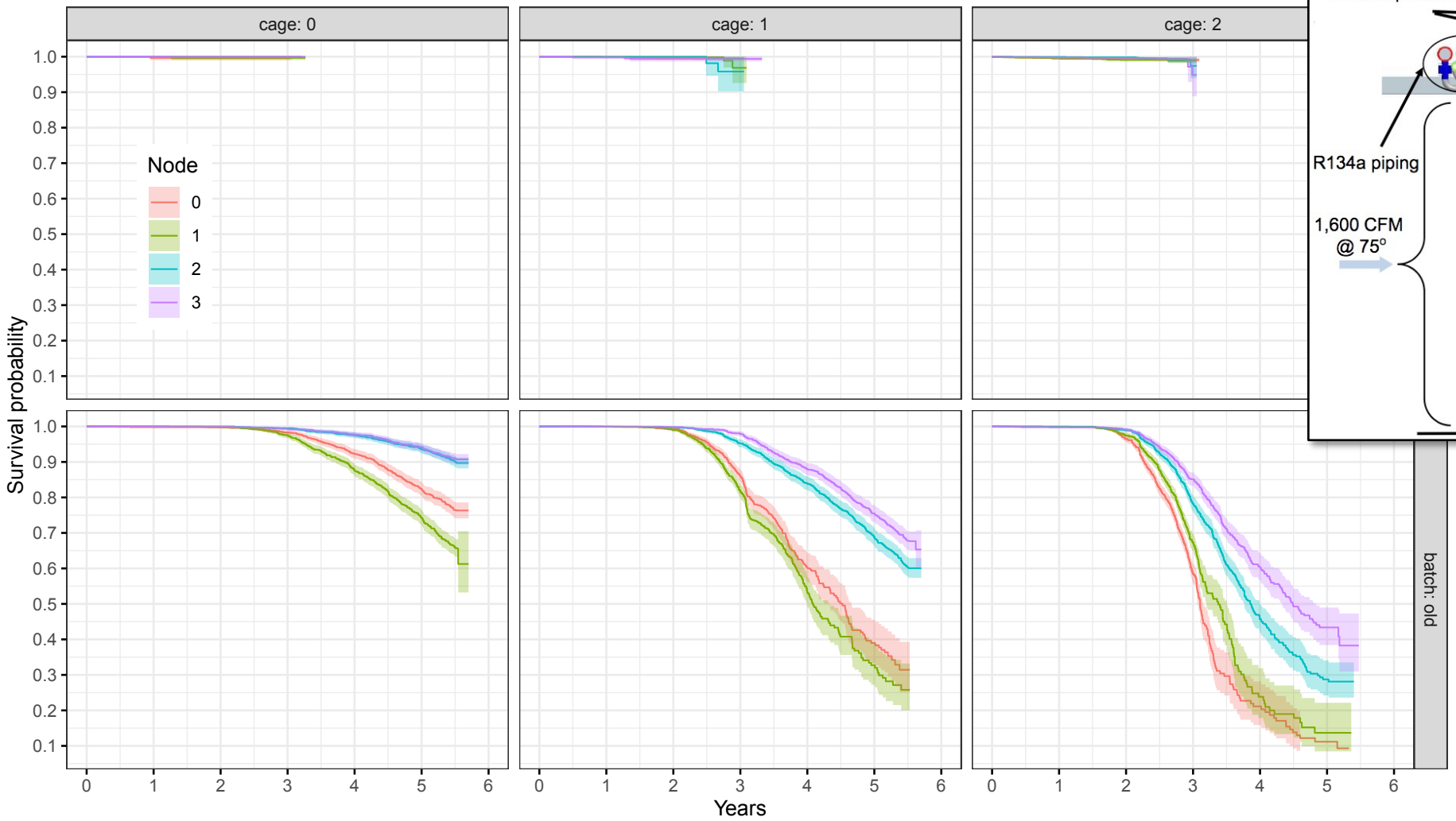


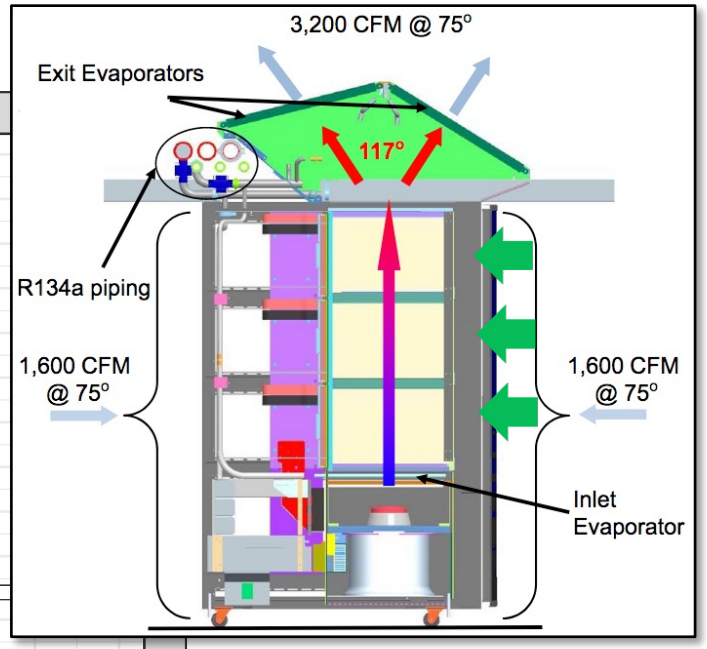
Old–New as Two Partitions: MTBF differs by 12x factor!

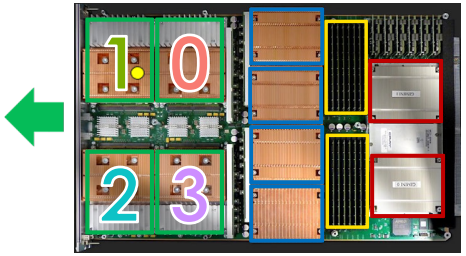
OAK RIDGE National Laboratory

G. Ostrouchov, D. Maxwell, R. Ashraf, C. Engelmann, M. Shankar, and James Rogers. **GPU Lifetimes on Titan Supercomputer: Survival Analysis and Reliability**. 33rd IEEE/ACM International Conference on High Performance Computing, Networking, Storage and Analysis (SC) 2020, pages 41:1-14, Atlanta, GA, USA, November 15-20, 2020. ISBN 9781728199986. DOI 10.1109/SC41405.2020.00045.

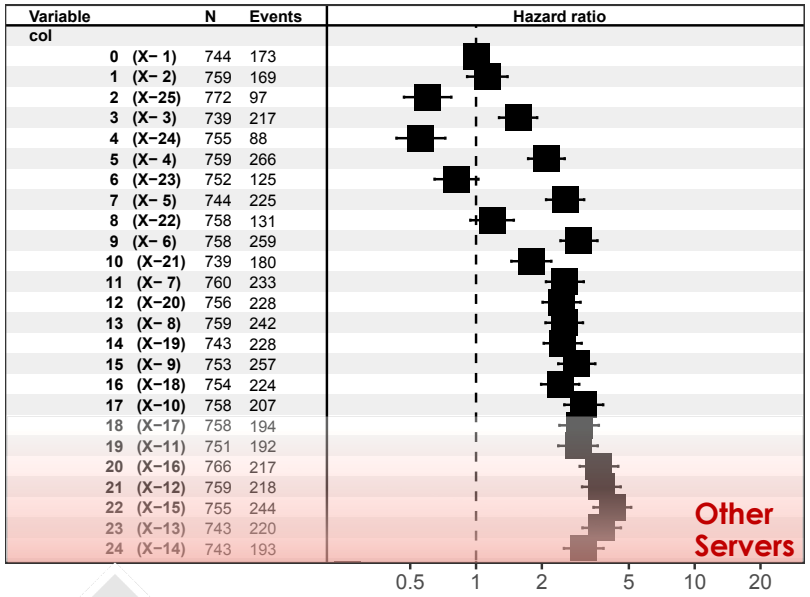
Cage and Node Effect Explainable by Airflow in Cabinet



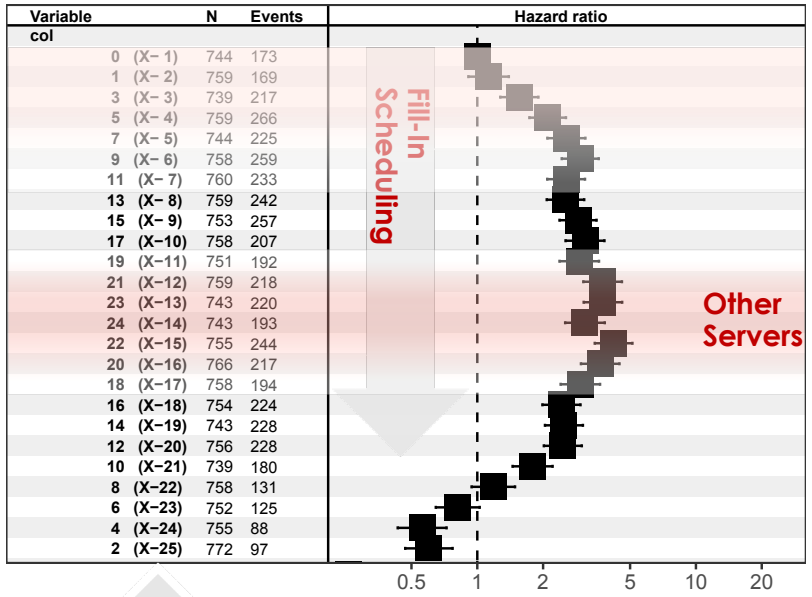




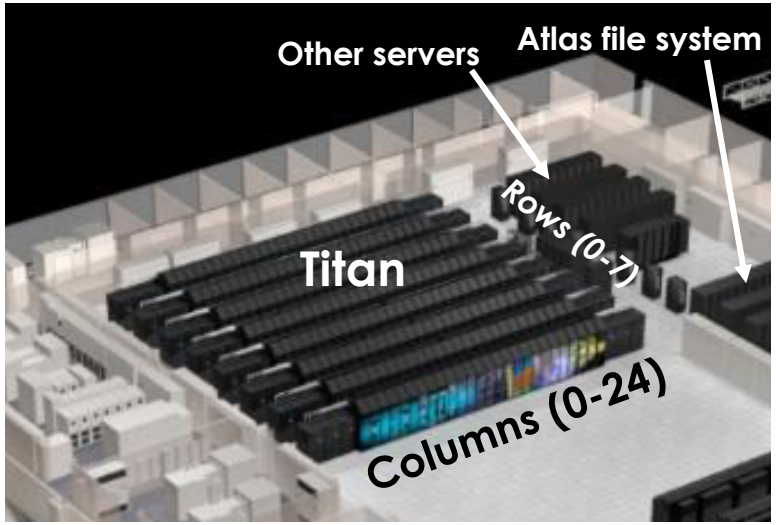
Fill-in Scheduling Effect Explainable via Torus Coordinate



Column Order



Torus Coordinate Order



DOE Early Career Award: Resilience Design Patterns

Novel Ideas:

- Design patterns that cover the hardware and software architecture aspects of resilience
- Methods and metrics to holistically evaluate and coordinate fault management
- Reusable programming templates for resilience portability
- Tools for trading off performance, resilience, and power consumption at design and run time

Impact:

- Enables the systematic improvement of resilience in extreme-scale systems
- Keeps applications running to a correct solution in a timely and efficient manner in spite of frequent faults, errors, and failures

Accomplishments:

- Resilience design pattern specification with taxonomy, survey, and pattern anatomy, classification, catalog and language
- GMRES solver with portable multi-resilience against process failures and data corruption
- Performance, reliability and availability models for 15 structural resilience design patterns

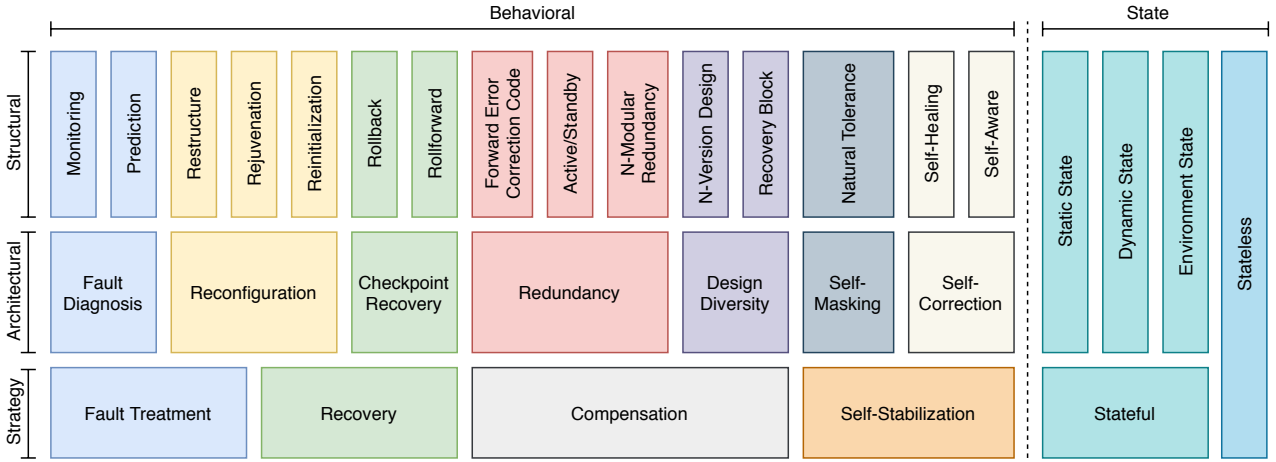
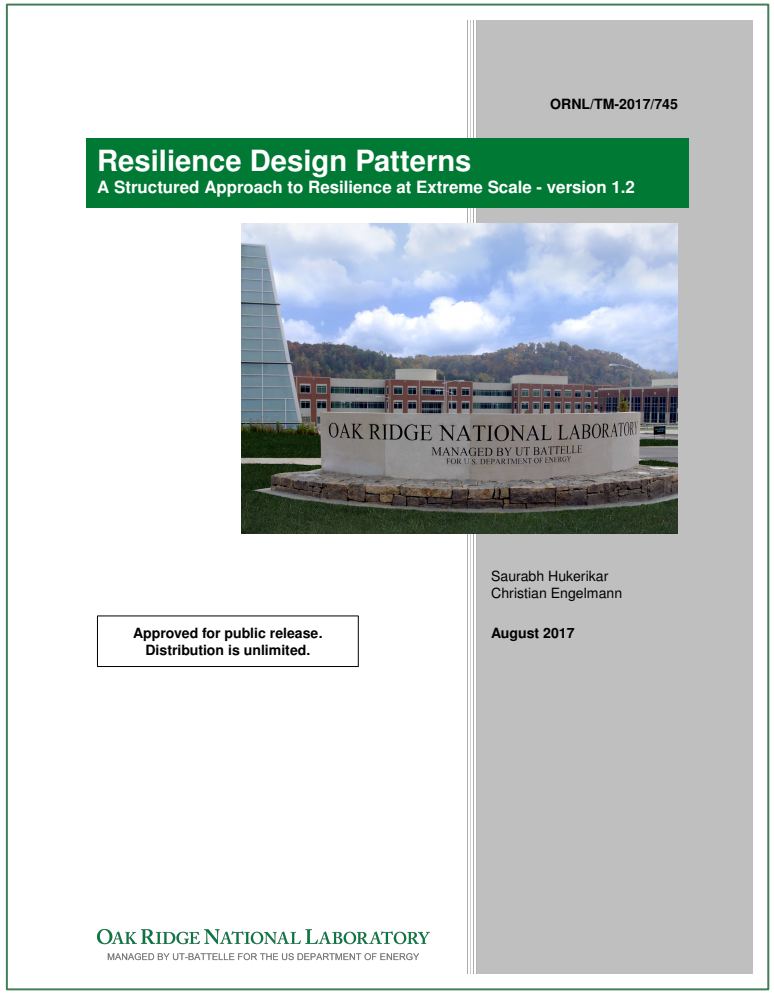


Figure: The 31 identified resilience design patterns

Resilience Design Patterns Specification

- Taxonomy of resilience terms and metrics
- Survey of resilience techniques
- Classification of resilience design patterns
- Catalog of resilience design patterns
  - Uses a pattern language to describe solutions
  - 4 strategy patterns, 7 architectural patterns, 15 structural patterns, and 5 state patterns
- Case studies using the design patterns
- A resilience design spaces framework
- Version 2.0 to be released soon


ORNL/TM-2017/745

Resilience Design Patterns
A Structured Approach to Resilience at Extreme Scale - version 1.2

Saurabh Hukerikar
Christian Engelmann

August 2017

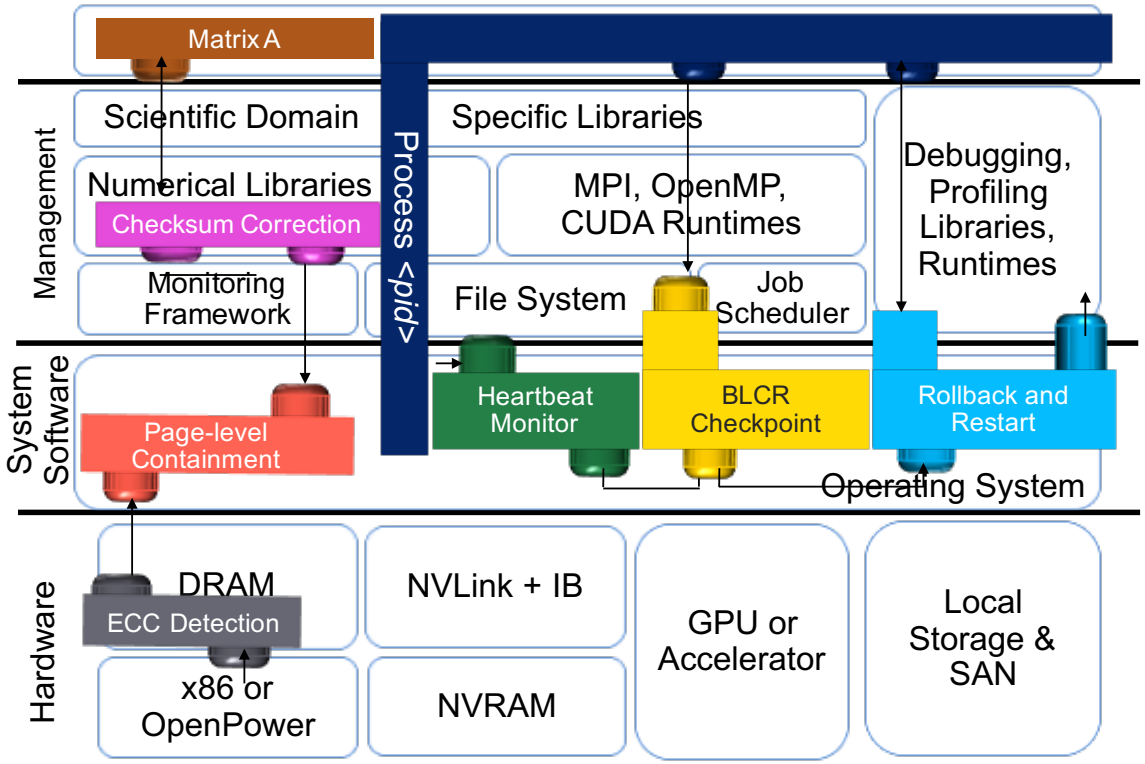
Approved for public release.
Distribution is unlimited.

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

Saurabh Hukerikar and Christian Engelmann. **Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale (Version 1.2)**. Technical Report, ORNL/TM-2017/745, Oak Ridge National Laboratory, Oak Ridge, TN, USA, August, 2017. DOI: 10.2172/1436045

Design Space Exploration for Resilience

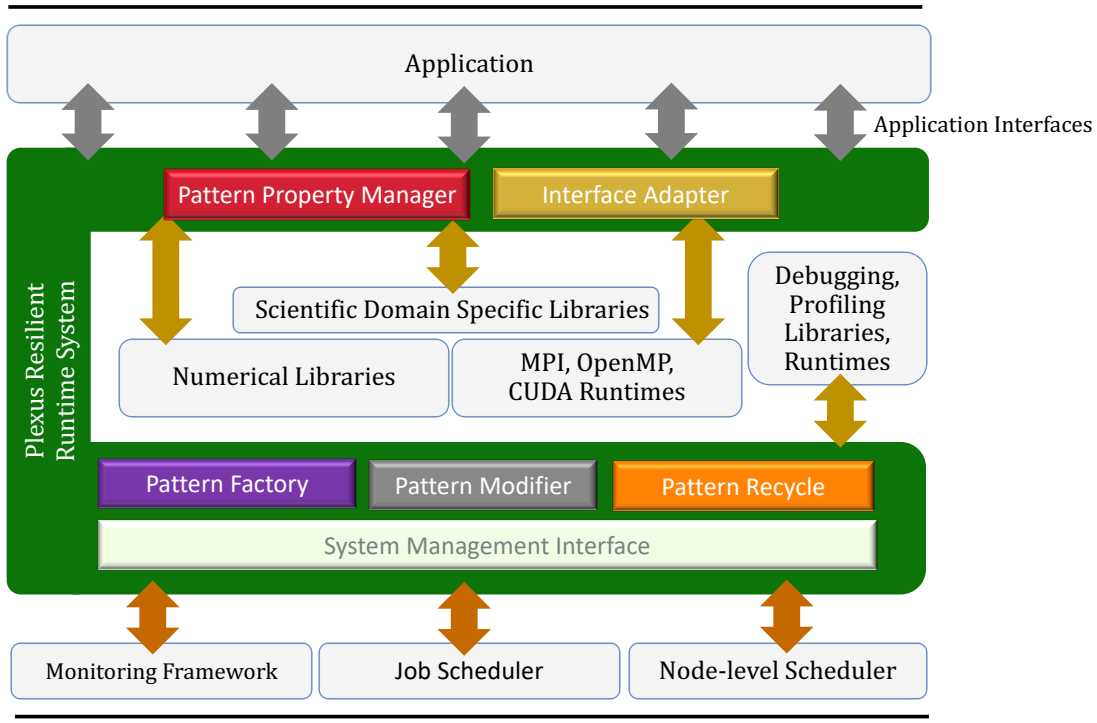
- Vertical and horizontal pattern compositions describe the resilience capabilities of a system
- Pattern coordination leverages beneficial and avoids counterproductive interactions
- Pattern composition optimizes the performance, resilience and power consumption trade-off



PLEXUS: A Pattern-Oriented Runtime System Architecture for Resilient Extreme-Scale High-Performance Computing Systems

- PLEXUS implements pattern instances to provide a resilient environment for HPC applications
- Offers strategies for the resilience patterns to be instantiated, modified and destroyed by the runtime based on policies to meet resiliency needs
- Prototype covers MPI process failures and transient data corruption for a GMRES solver.

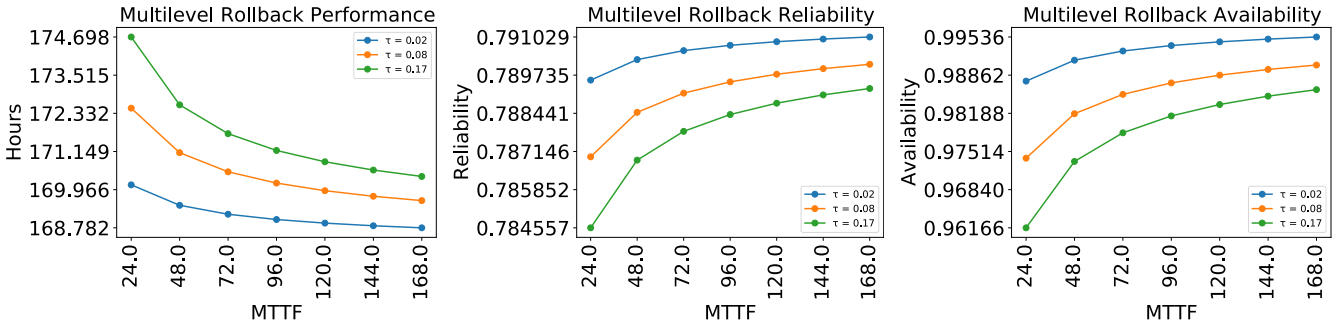
S. Hukerikar and C. Engelmann. **PLEXUS: A Pattern-Oriented Runtime System Architecture for Resilient Extreme-Scale High-Performance Computing Systems**. *25th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC) 2020*, Perth, Australia, December 1-4, 2020.



Architecture of the Plexus resilient runtime system, interfacing with programming model runtimes, libraries, system monitoring and job and resource management.

RDPM: An Extensible Tool for Resilience Design Patterns Modeling

- Permits exploring the performance, reliability, and availability design space of extreme-scale supercomputers
- Offers customization of design parameters to optimize performance, reliability, and availability
- Allows the investigation of trade-offs for combining multiple individual resilience solutions
- Enables providing the most coverage against faults, errors and failures while minimizing the impact on performance



(a) Performance (b) Reliability (c) Availability

The performance, reliability, and availability of multi-level rollback (e.g., accelerator-level and application-level checkpoint/restart) modeled by the RDPM software tool with a varying system mean-time-to-failure (MTTF) of 24-168 hours (1-7 days), 80% of the computation offloaded to the accelerator and protected by both levels, a 1 second checkpoint/restart time at the accelerator level and a 1, 5 or 10 minute checkpoint/restart time at the application level.

M. Kumar and C. Engelmann. **RDPM: An Extensible Tool for Resilience Design Patterns Modeling**. In *Lecture Notes in Computer Science: Proceedings of the 27th European Conference on Parallel and Distributed Computing (Euro-Par) 2021 Workshops: 14th Workshop on Resiliency in High Performance Computing (Resilience) in Clusters, Clouds, and Grids*, August, 2021.

Future Research and Development Needs (1/2)

- Resilience needs to become an integral part of the HPC hardware/software ecosystem through codesign
- The burden for resilience should be on the system by design and not on the operator or user as an afterthought
- Future smart HPC systems employ coordinated cross-layer and adaptive resilience solutions to:
  - Offer efficient error and failure masking, recovery, and avoidance at the appropriate hardware or software component and compute or data granularity
  - Handle errors and failures in specific components and granularities where it is most appropriate to do so and in coordination with the rest of the system

Future Research and Development Needs (2/2)

- In the short term:
  - Portable system/center monitoring and analysis solutions to enable identification of emerging reliability issues and their root causes
  - Low-overhead software mitigation techniques (beyond global checkpoint/restart) to create a better resilience toolbox that can be used when needed
- In the long term:
  - Autonomous resource management that considers the system/facility state and the involved performance, resilience and power consumption trade-offs
  - Autonomous adaptation of systems and facilities to emerging reliability issues
  - Machine-in-the-loop operational intelligence for systems and centers (OODA loop to improve productivity and lower costs)

Questions?