Fine-grained Automated Failure Management on Extreme-Scale GPU Systems

2025 Feb 20

Balazs Gerofi <balazs.gerofi@intel.com>

Multicore World 2025, Christchurch, New Zealand

intel®

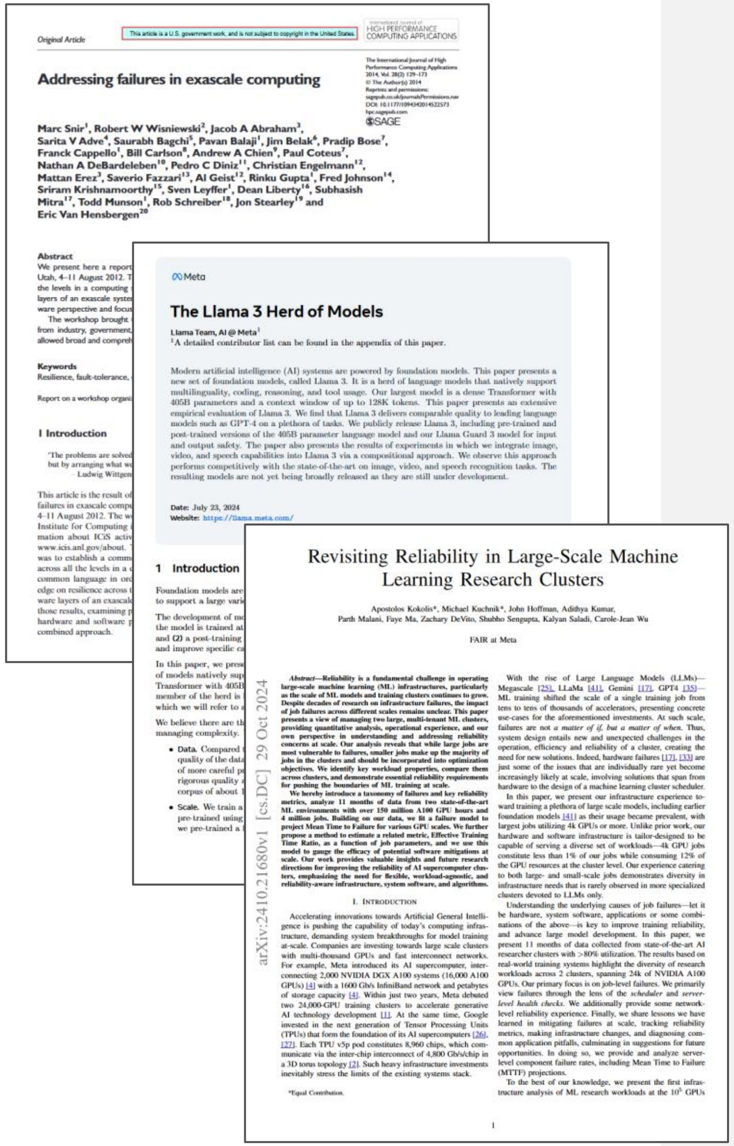
Legal Disclaimer

Agenda

- Motivation
- Background
  - Aurora Overview
  - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

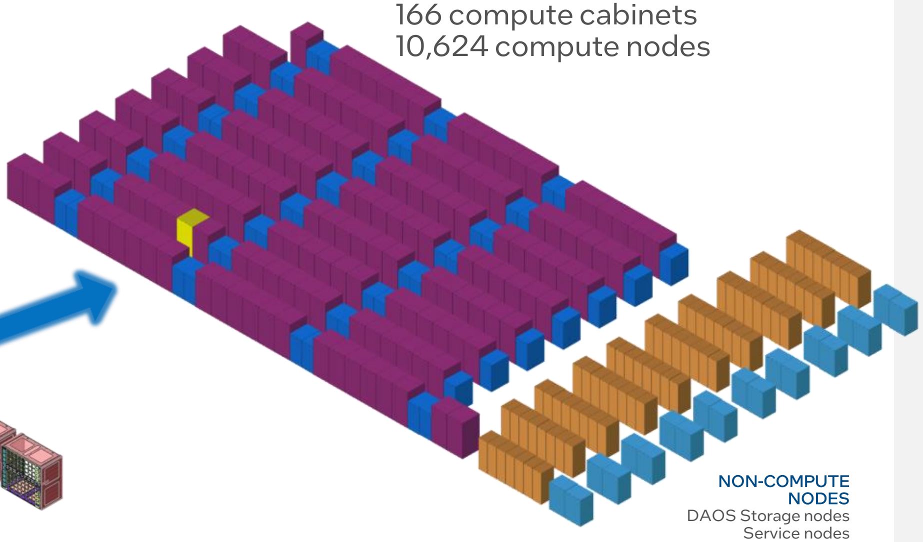
Motivation

- Failures in leadership class accelerated HPC and AI systems have become the norm rather than the exception
  - This was anticipated a decade ago in the HPC community…
- As systems continue to scale in size, the frequency of failures on the entire system is expected to increase
- Tightly coupled parallel workloads (e.g., HPC modeling/simulation and AI training/fine-tuning) are highly sensitive to failures
  - A single "*lemon node*" can ruin the run
- To ensure efficient deployment and operation, automated failure management is essential

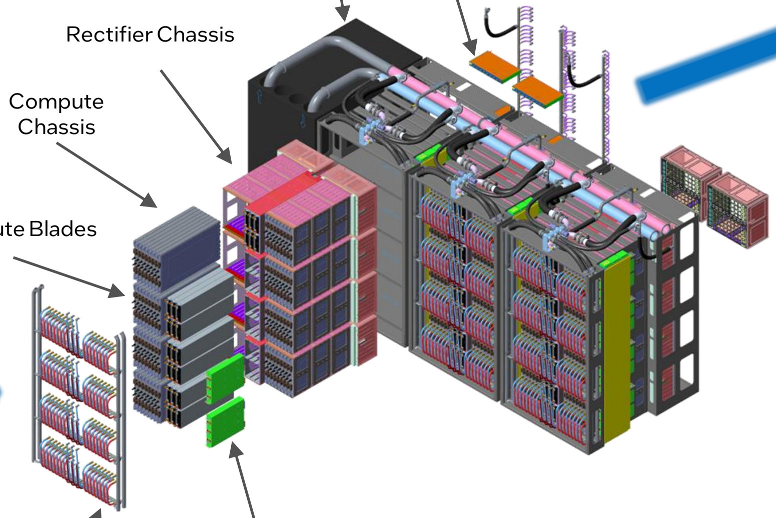


Aurora System Architecture

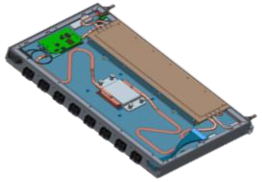
COMPUTE SYSTEM
166 compute cabinets
10,624 compute nodes



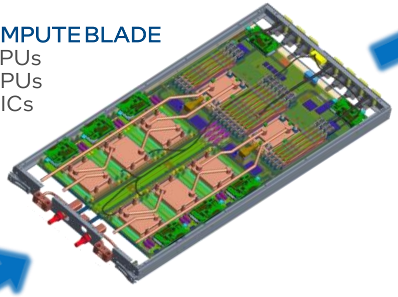
COMPUTE CABINET
64 compute blades

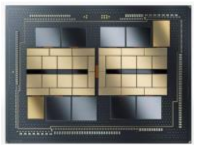


SWITCH BLADE
1 Slingshot switch
DragonFly topology



COMPUTE BLADE
2 CPUs
6 GPUs
8 NICs





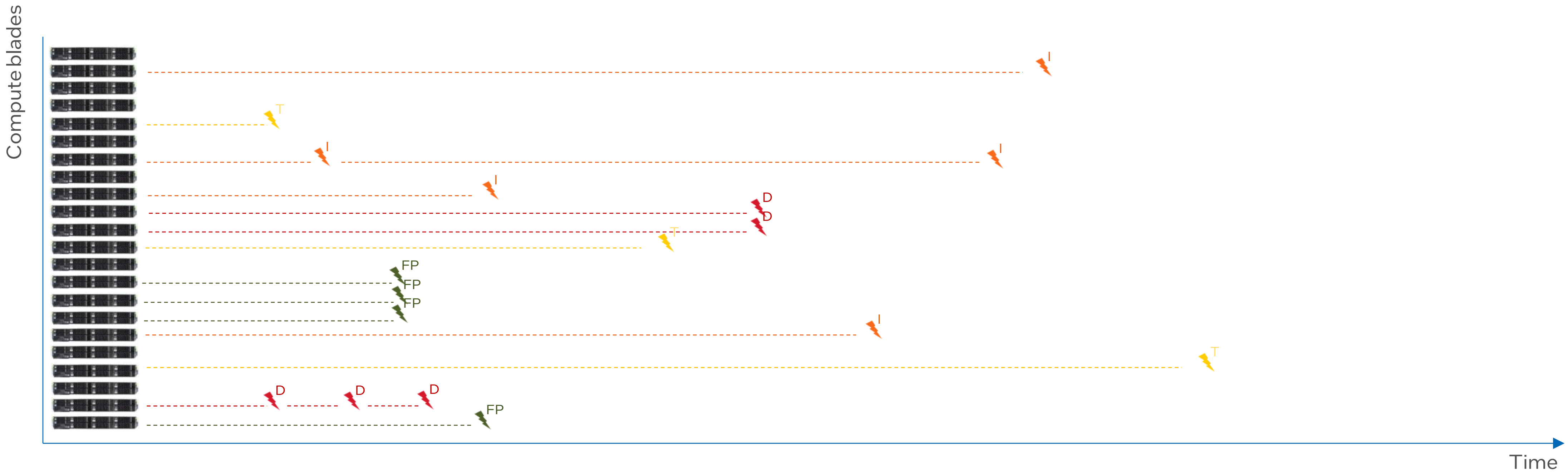
GPU
Ponte Vecchio
2-Tile Configuration
8x 16GB HBM stacks

CPU
SPR+HBM SoC
52 Xeon cores
4x 16GB HBM stacks
8x DDR5 channels

AURORA SYSTEM
10,624 compute nodes
63,744 GPUs
21,248 CPUs
84,992 NICs
84,992 IVOCs (PSU)
5,312 Switches
169,984 DDR DIMMs
594,944 HBM Stacks

Over ~345,000 replaceable/repairable parts in the system!

Hardware Failures in Large Scale Systems



(D) Failures due to design bugs which fail on every instance of component <X> under specific conditions (e.g., MERT accumulator overflow bug in PVC)

(I) Failures due to intermittent faults

- Aged/degraded components, systematic issues due to marginalities (could also be deficiencies in design, validation, etc.), manufacturing defects

(T) Transient/random failures due to particle/EM radiation/cosmic rays

- These will create a "background" noise of failures that should be distributed evenly across the system

(FP) Software, networking, or correlated faults and other errors that could create failures that look like hardware component issues

Acknowledgments: Lance Cheney and Gustavo Espinosa @ Intel

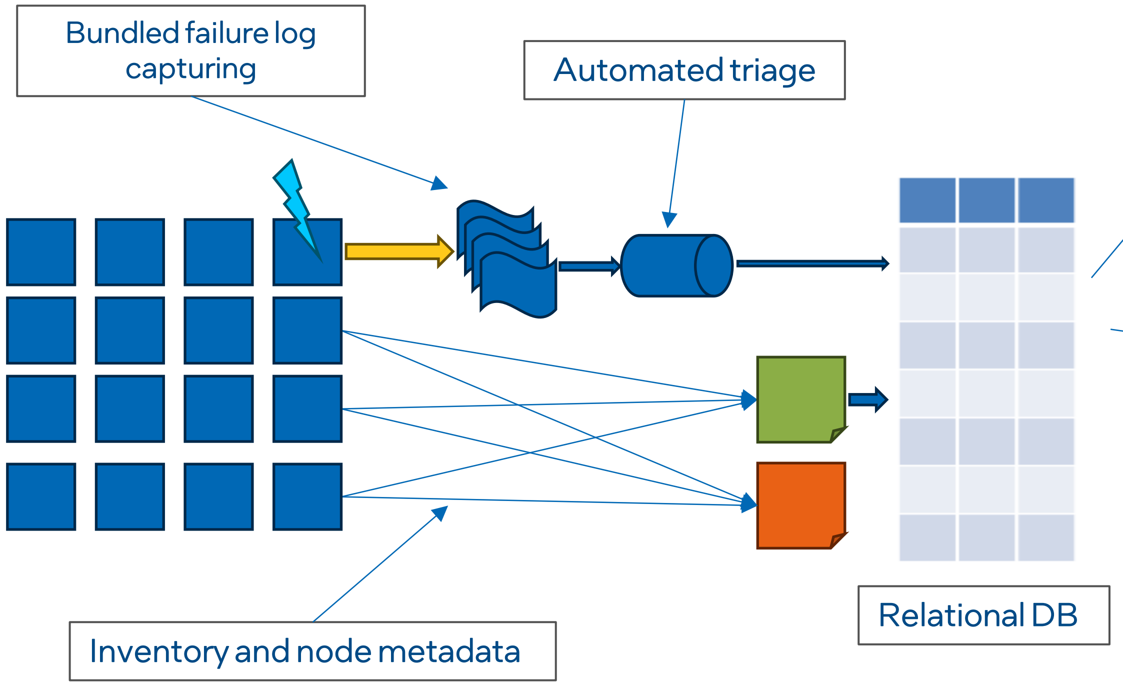
Key Observations and Problem Statement

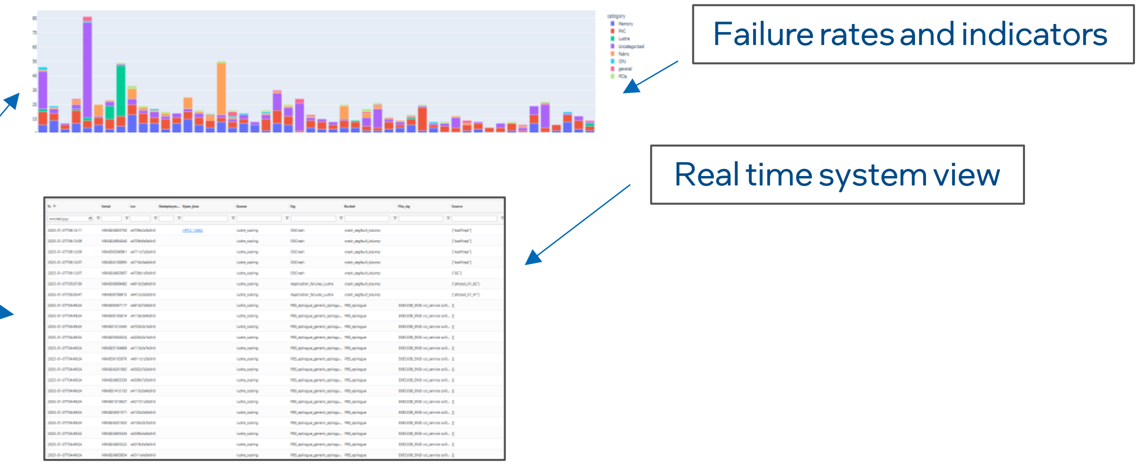
- How to distinguish *intermittent* and *transient* failures?
- How to handle first strikes?
  - Replacing every component on first strikes is *impractical*
- Probability of an intermittent error occurring on the same component twice is extremely small
  - Indication of defects?
- Need to understand *reoccurrence rates* and the *statistical properties of durations between strikes*
  - These are specific to failure modes
- Failure history needs to be captured *in context*
  - Firmware/software versions, external conditions have impacts
- Need for automated failure categorization
- Automated failure servicing/management?

Agenda

- Motivation
- Background
  - Aurora Overview
  - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

StabilityDB Infrastructure Architectural Overview





StabilityDB is an integrated timeseries based meta-database on compute nodes in the system, history of:

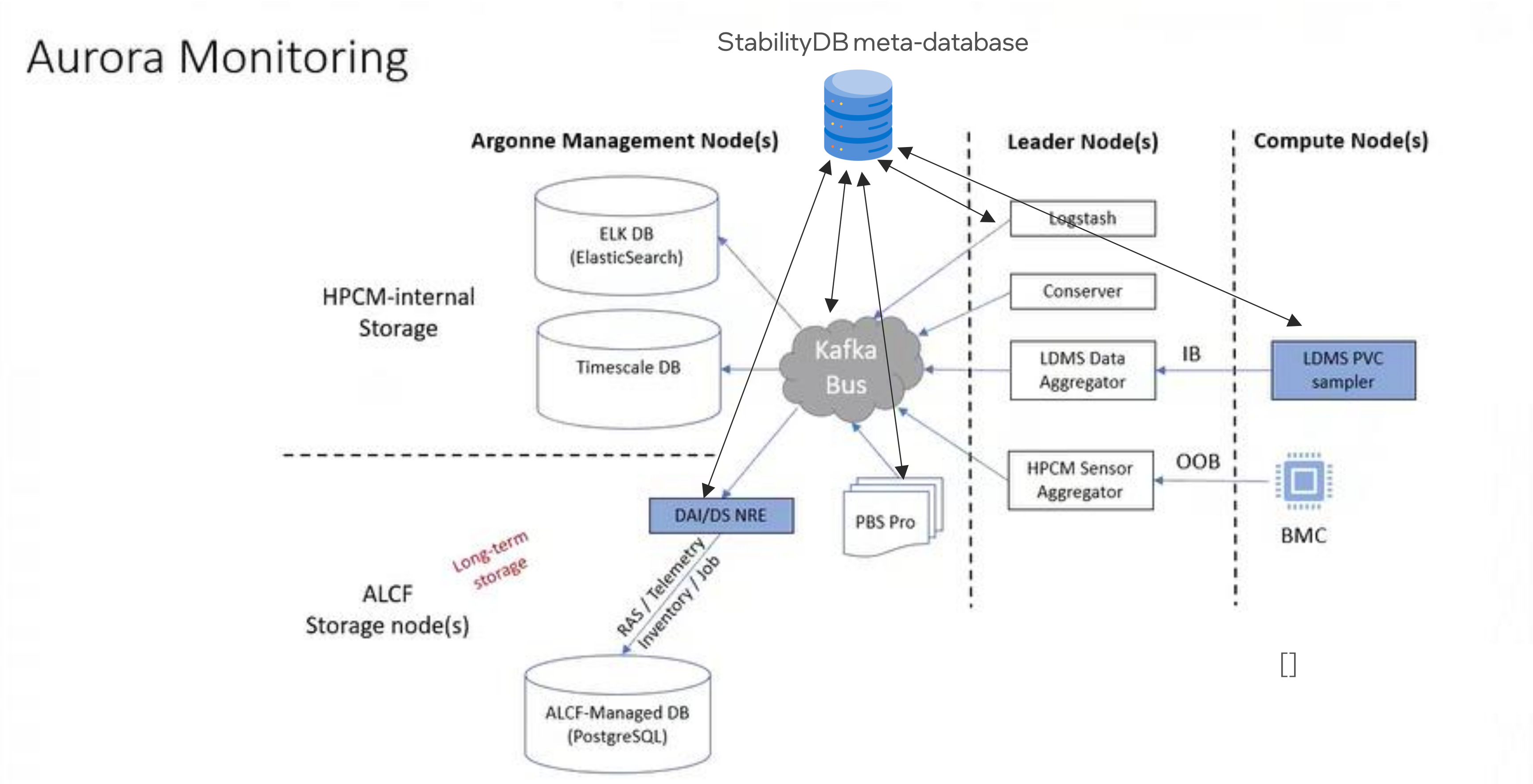
- Failures
- Inventory/component serials
- Versioning (firmware/software/etc.)
- Workloads
- Resource manager state, e.g., queue changes, etc.

Primary insight was "essential data in context", with goals of:

- Easy analysis
- Data driven guidance for bring-up/debug

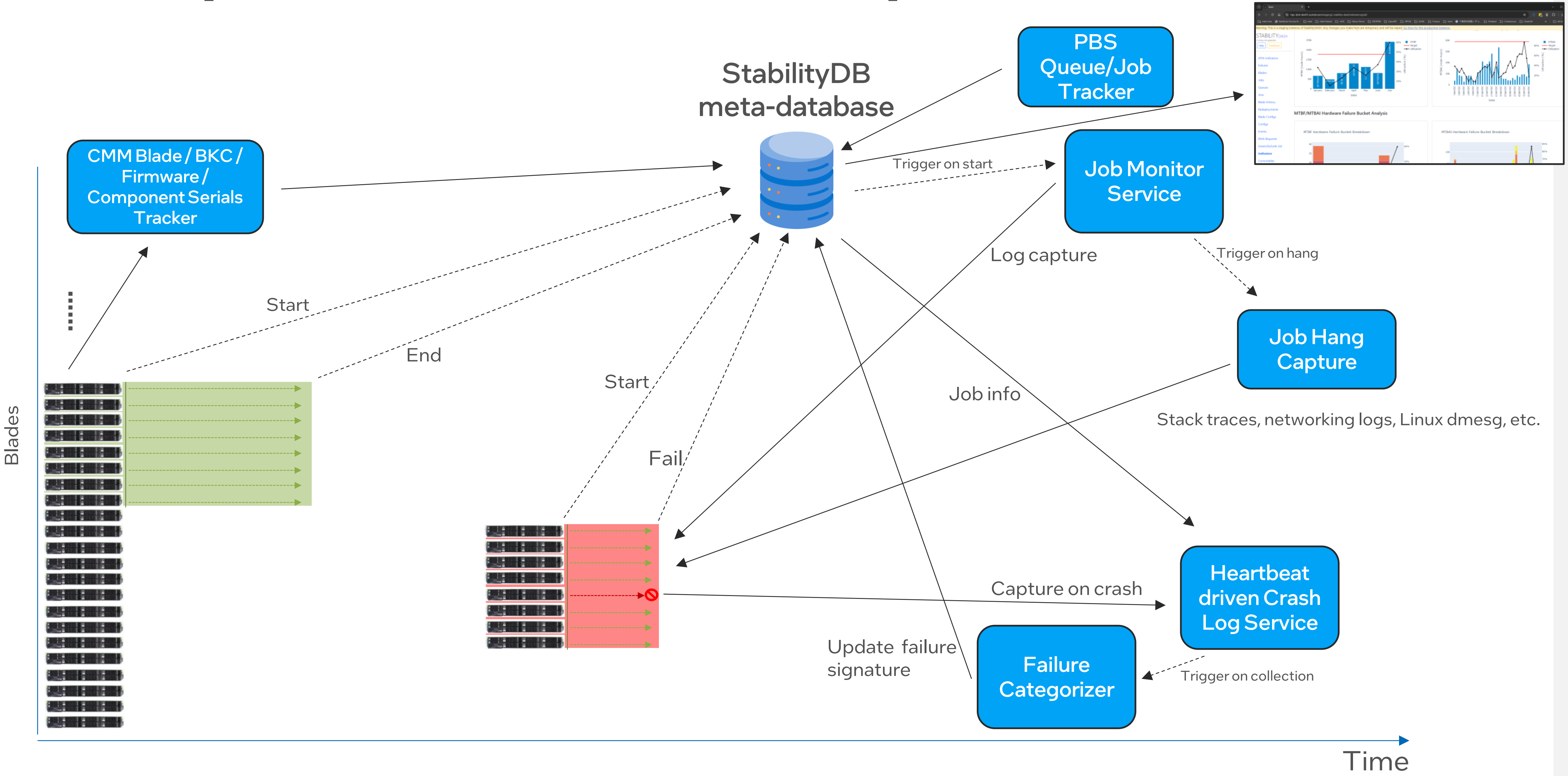
- MTBF/MTBAI analysis
- Failure stats/breakdowns with workload association
- Failure stats with component/FW versioning association
- Inventory/part replacement tracking and projections
- Automated failure mitigation

Aurora: Interaction with Cluster Management/Telemetry



- Plugs into standard telemetry/event data streams:
    - E.g., Kafka and RabbitMQ
- Interacts with cluster management software components (e.g., HPE HPCM node management)
- Interacts with batch scheduler (e.g., PBS)

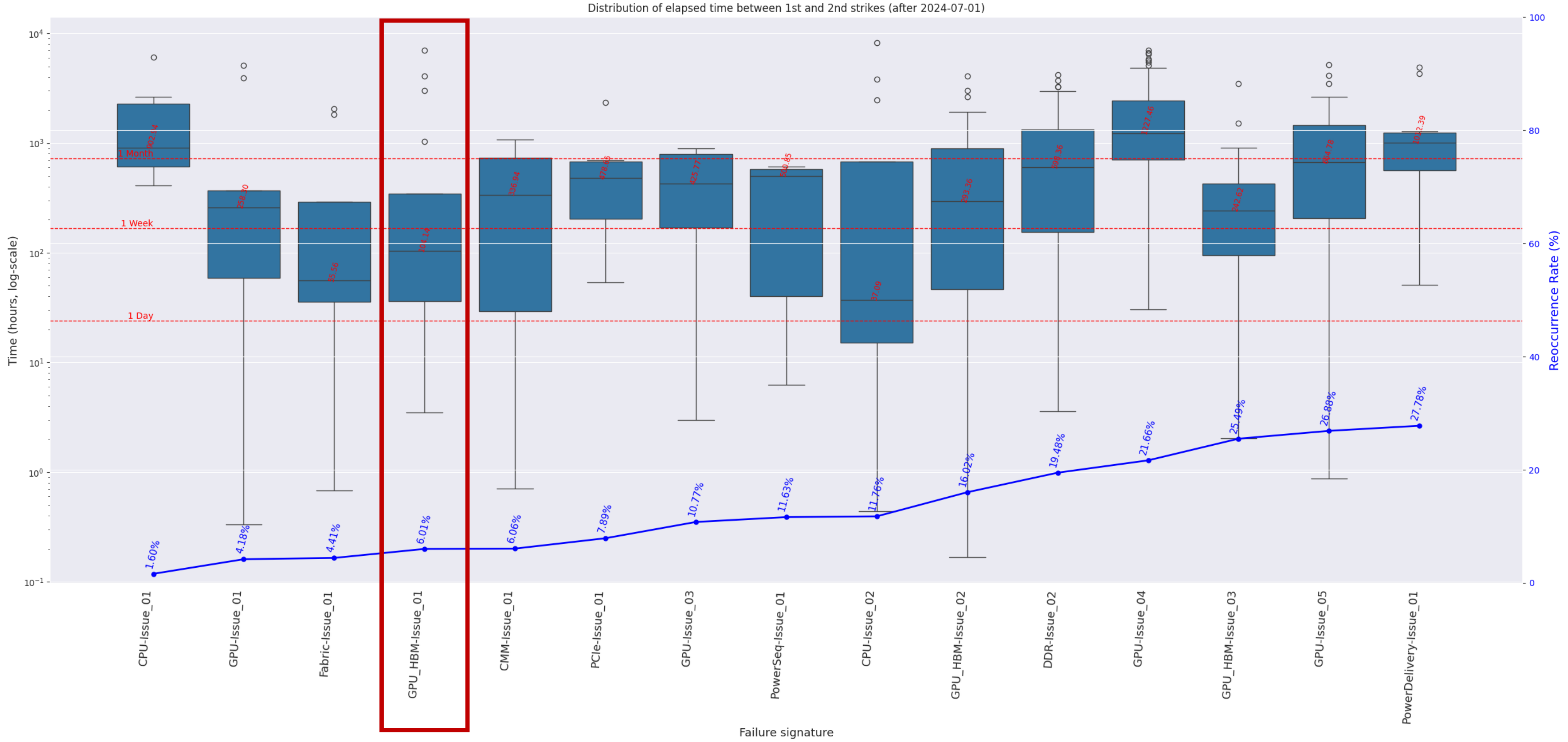
StabilityDB Backend Functional Components



Agenda

- Motivation
- Background
  - Aurora Overview
  - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

Distribution of Elapsed Time between 1st and 2nd Strikes



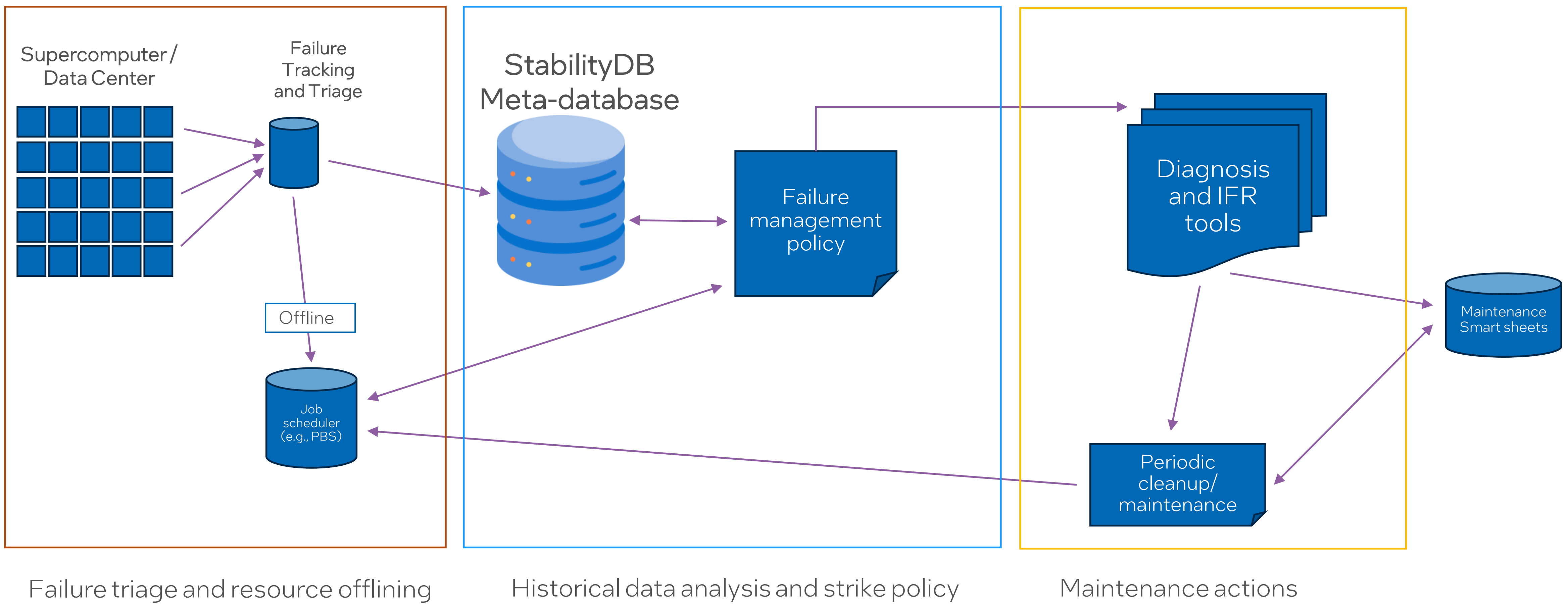
- Low reoccurrence rate
- Higher reoccurrence rate but long elapsed time between strikes

Fine-Grained Multi-Strike Policies

Updated_ts	Sig	Live	Criteria	Strike_1	Strike_2	Strike_3	Strike_4	Strike_5
2024-12-08 13:02:41	fatal_7_multiple_quads	1	single chip	HOST_COLD_RESET,REDEPLOY_STABILITYDB...	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	fatal_7_others	1	single chip	HOST_COLD_RESET,REDEPLOY_STABILITYDB...	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	fpu_fatal_singlesubslice_error	1	single source	DSS_SWAP,MONITOR	DSS_SWAP,MONITOR	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	general_multibank_correctable_error	1	single chip	HOST_COLD_RESET,REDEPLOY_STABILITYDB...	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR
2024-12-08 13:02:42	general_multibank_error	0	single chip	HOST_COLD_RESET,REDEPLOY_STABILITYDB...	GT IFR	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	general_multisubslice_correctable_error	1	single chip	HOST_COLD_RESET,REDEPLOY_STABILITYDB...	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR	REDEPLOY_STABILITYDB,MONITOR
2024-12-08 13:02:42	general_multisubslice_error	1	single chip	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	general_singlesubslice_error	0	single source	DSS_SWAP,MONITOR	DSS_SWAP,MONITOR	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	grf_fatal_singlesubslice_error	1	single source	DSS_SWAP,MONITOR	DSS_SWAP,MONITOR	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	grf_singlesubslice_correctable_error	1	single source	DSS_SWAP,MONITOR	DSS_SWAP,MONITOR	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	GPU_HBM-Issue_01	1	single source	HBM_IFR,PVC_COLD_RESET,MONITOR	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	hbm_controller_multi_errors	0	single chip	HBM_IFR,PVC_COLD_RESET,MONITOR	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	hbm_data_errors	1	single source	HBM_IFR,MONITOR	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP
2024-12-08 13:02:42	hbm_training_errors	1	single source	HBM_IFR,PVC_COLD_RESET,MONITOR	RMA_BTK	CONTACT_INTEL_REP	CONTACT_INTEL_REP	CONTACT_INTEL_REP

- Per failure mode fine-grained description
- Policies must be based on statistical information, for that we need a meta-database tracking those metrics
- Defines failure management automation actions

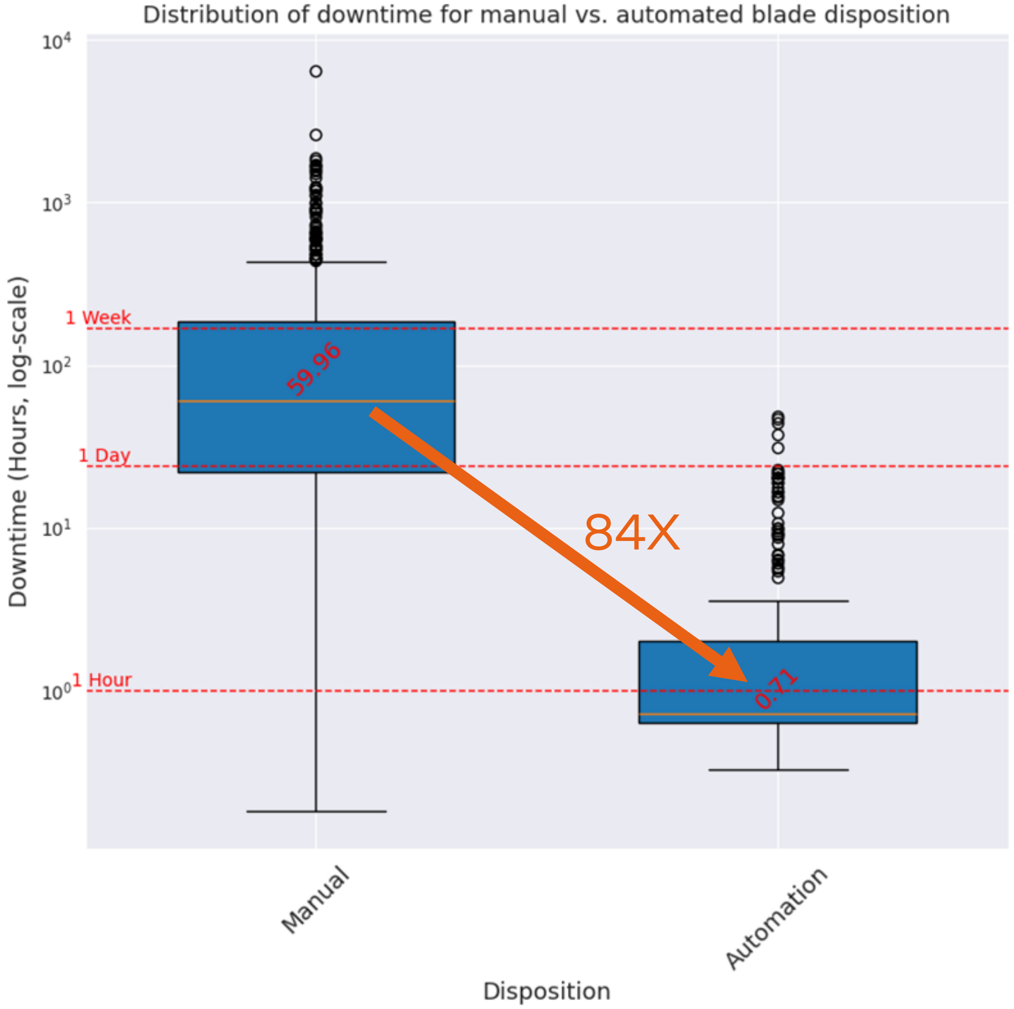
Automated Failure Management Components

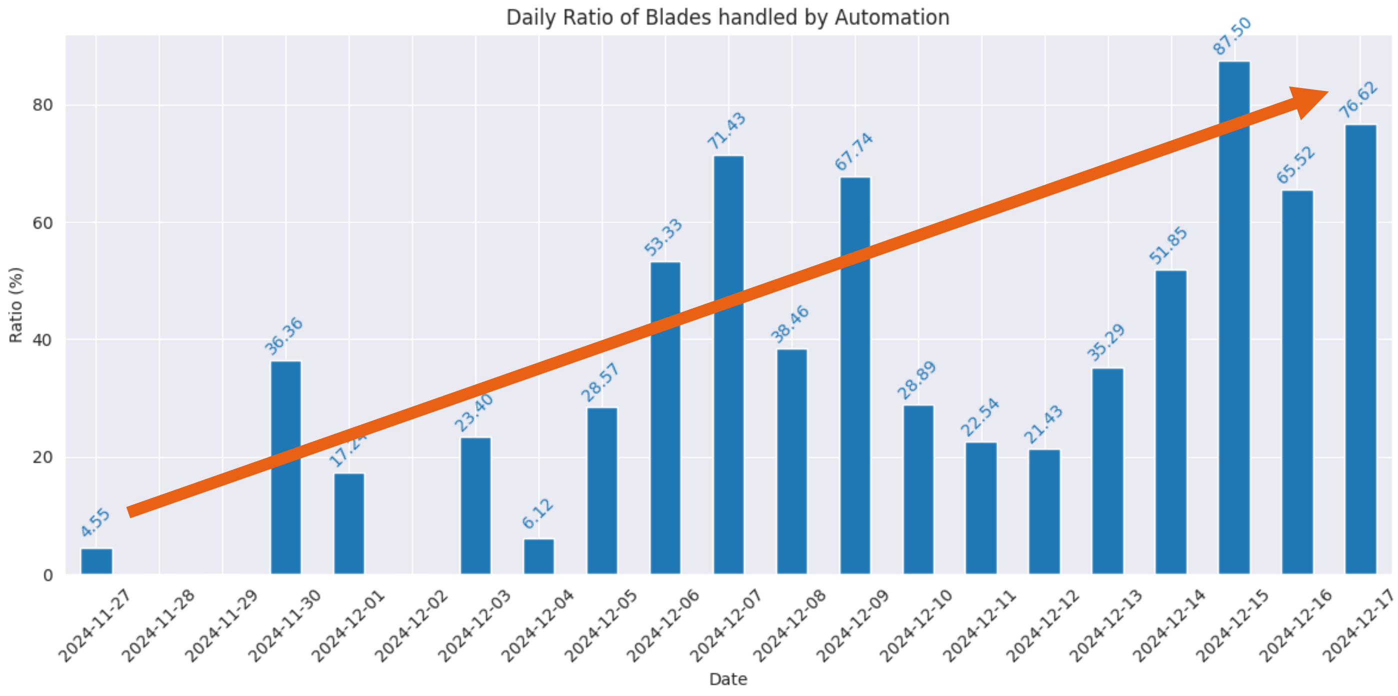


Agenda

- Motivation
- Background
  - Aurora Overview
  - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

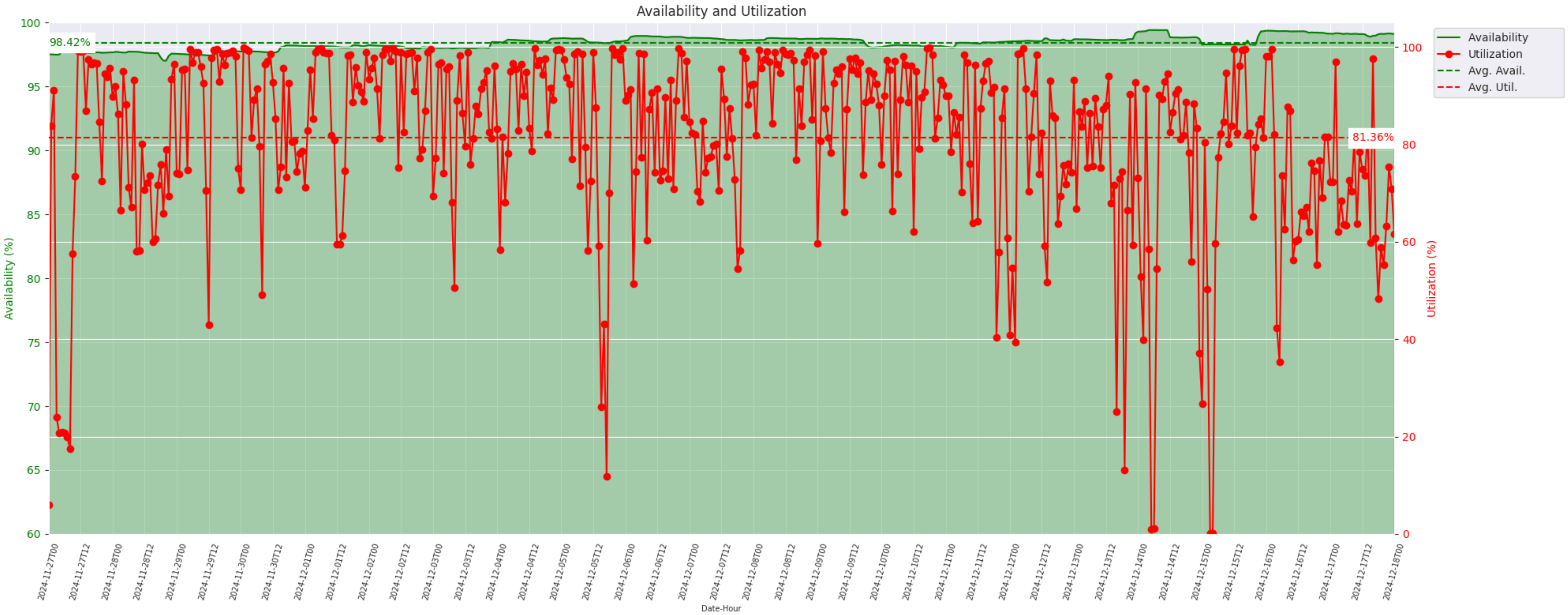
Automated Failure Management Results (AT-S)





- Excludes large correlated events (e.g., Lustre mount failures, rack EPO events, etc.)
- Median time spent on addressing issues is 2 orders of magnitude lower!
- Automation's ratio steadily increasing for blade dispositions

Availability and Utilization during AT-S



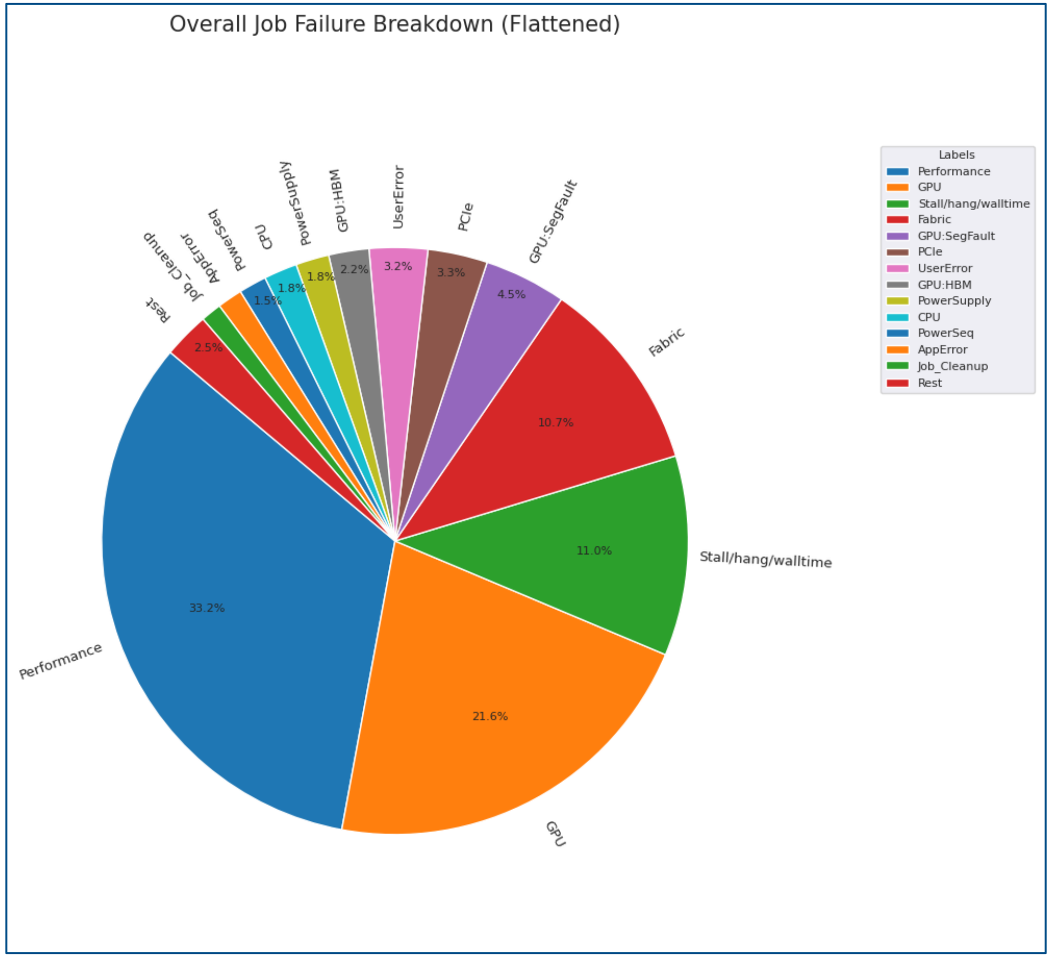
- Requirement of 95% availability in average

First and Second Strikes Month over Month

- System stabilizing via decreasing 1st and 2nd counts for most failure signatures
- Constant rate of CEs on DDR and HBM

Job Failure Breakdown during AT-S vs. Meta

*The Llama3 herd of models



Very similar overall failure profiles!

This equals to an MTBAI = 3.7 H @ 16K GPUs
Scaling this linearly to Aurora scale:
MTBAI = 55 min @ 64K GPUs

Component	Category	Interruption Count	% of Interruptions
Faulty GPU	GPU	148	30.1%
GPU HBM3 Memory	GPU	72	17.2%
Software Bug	Dependency	54	12.9%
Network Switch/Cable	Network	35	8.4%
Host Maintenance	Unplanned Maintenance	32	7.6%
GPU SRAM Memory	GPU	19	4.5%
GPU System Processor	GPU	17	4.1%
NIC	Host	7	1.7%
NCCL Watchdog Timeouts	Unknown	7	1.7%
Silent Data Corruption	GPU	6	1.4%
GPU Thermal Interface + Sensor	GPU	6	1.4%
SSD	Host	3	0.7%
Power Supply	Host	3	0.7%
Server Chassis	Host	2	0.5%
IO Expansion Board	Host	2	0.5%
Dependency	Dependency	2	0.5%
CPU	Host	2	0.5%
System Memory	Host	2	0.5%

Table 5 Root-cause categorization of unexpected interruptions during a 54-day period of Llama 3 405B pre-training. About 78% of unexpected interruptions were attributed to confirmed or suspected hardware issues.

Supercomputer/Data Center Digital Twins?

- “Digital twins provide living digital models of physical systems that enable **data-driven analysis** and application of artificial intelligence to better manage the datacenter and **drive efficiency** for sustainability.” [1]

- “Historically, data center management has been split into silos that each focus on one aspect.. as a result, ... different areas can miss the bigger picture. Digital twins help to **centralize data** from across different areas of concern **into a shared environment**” [2]

- Areas of potential:
  - Design: placing new servers, increasing density, improving thermal performance, etc.
  - Construction: streamlining construction, **reducing waste**, etc.
  - Operations: automating data center processes, **efficient maintenance and repairs**, etc.
  - Planning: ensuring compliance with data twins, understanding material impact, etc.

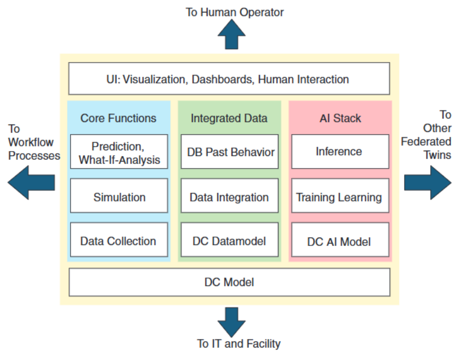
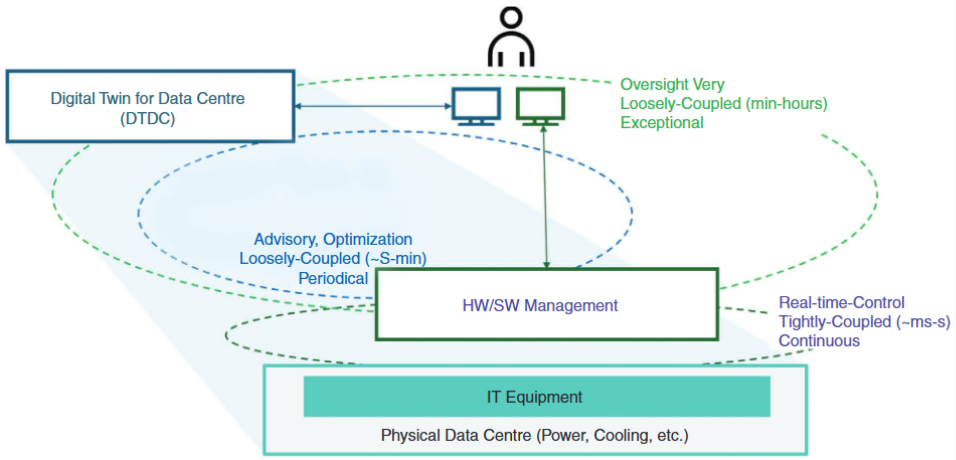


FIGURE 5. Data center digital twin software architecture. DB: database.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10687340



https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10687340

[1] J. Athavale et al., "Digital Twins for Data Centers," in Computer, vol. 57, no. 10, pp. 151-158, Oct. 2024, doi: 10.1109/MC.2024.3436945.

[2] https://venturebeat.com/ai/19-ways-digital-twins-improve-data-center-sustainability/

Conclusions and Outlook

- As large scale systems grow in size intermittent failures become more prevalent
- Efficient operation requires automated failure management
- Fine-grained multi-strike management policy
- Key is data in context that enables real-time decision making
- Outlook:
  - Predictive (AI?) failure avoidance
  - Continuous fleet scanning
  - Standardized failure reporting across components?

Acknowledgments/Contributors

- Yoni Levitt
- Richard Barella
- Erik Adames
- Leobardo Rountree
- Sam Zeltner
- Spurthi Lokeshappa
- Tom Musta
- Damon Millar
- Aravind Balasubramanian
- Patrick Steinbrecher
- Arjun Kripanidhi
- Aakash Patel
- Neha Gupta
- Kevin Canada
- Ky Merril
- Brian Holland
- Sucheta Raghunanda
- Ben Allen (ANL)
- Peter Upton (ANL)
- Doug Waldron (ANL)

Thank you for your attention!
Questions?