

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



Legal Disclaimer

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies may require enabled hardware, software or service activation.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more on the [Performance Index](#) site.

Your costs and results may vary.

"Conflict-free" refers to products, suppliers, supply chains, smelters, and refiners that, based on our due diligence, do not contain or source tantalum, tin, tungsten or gold (referred to as "conflict minerals" by the U.S. Securities and Exchange Commission) that directly or indirectly finance or benefit armed groups in the Democratic Republic of the Congo or adjoining countries.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Altering clock frequency or voltage may void any product warranties and reduce stability, security, performance, and life of the processor and other components. Check with system and component manufacturers for details.

Results have been estimated or simulated.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting the [Intel Resource and Document Center](#).

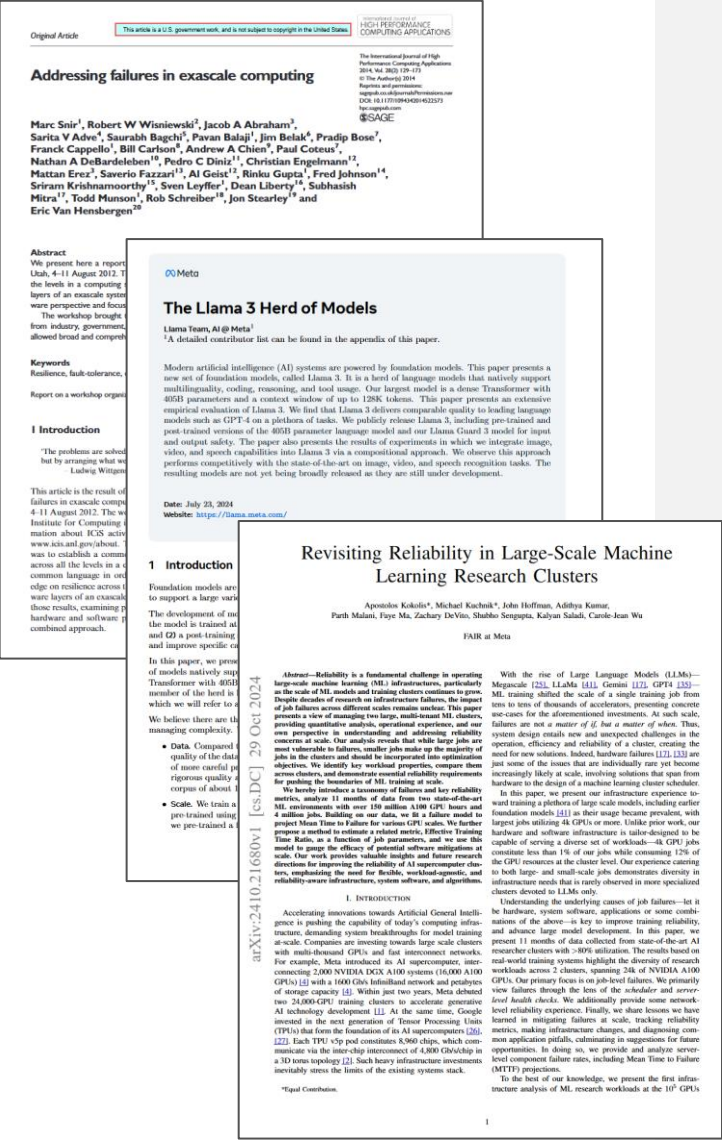
© 2025 Intel Corporation. Intel, the Intel logo, OpenVINO and the OpenVINO logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Agenda

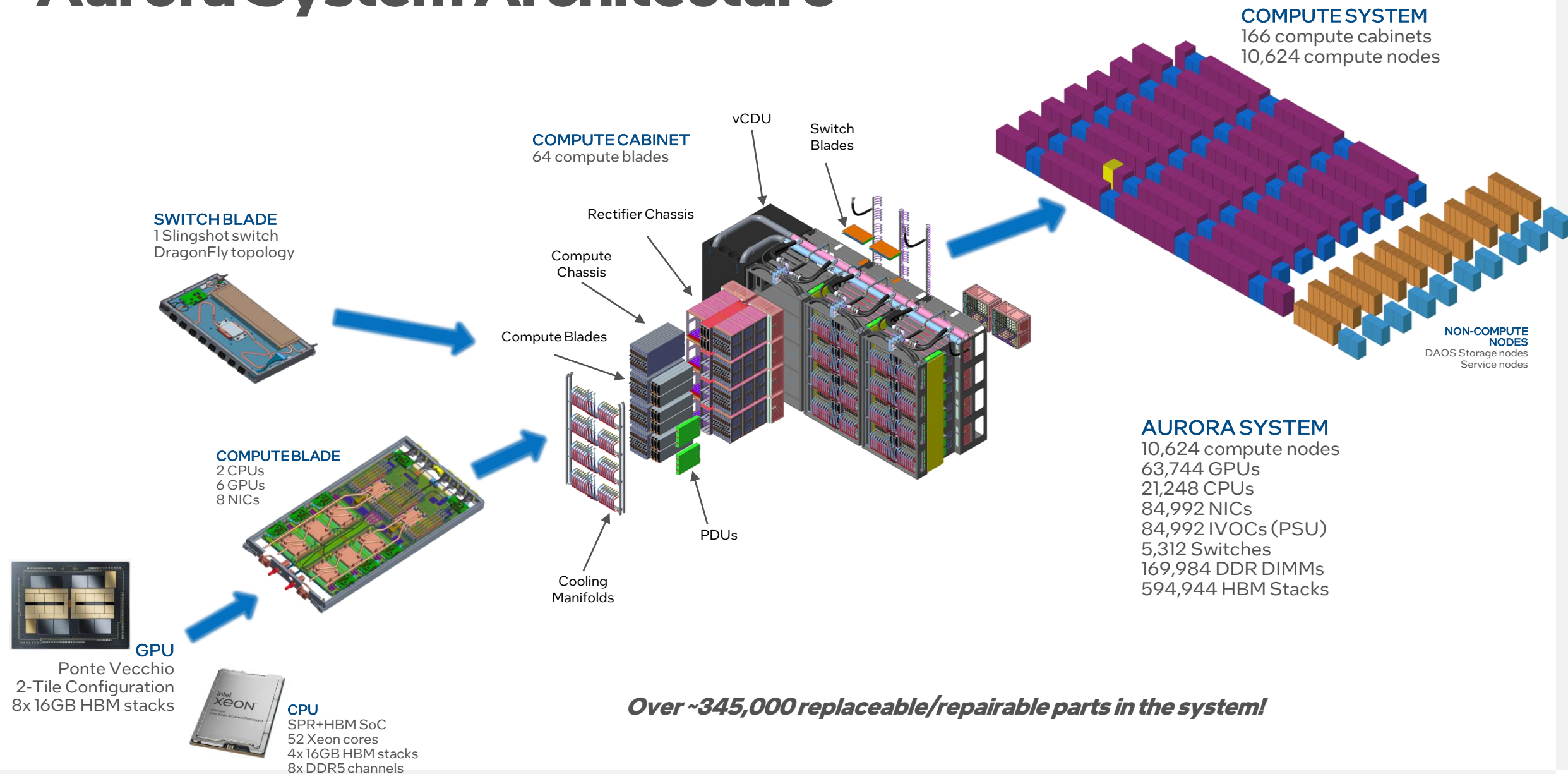
- Motivation
- Background
 - Aurora Overview
 - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

Motivation

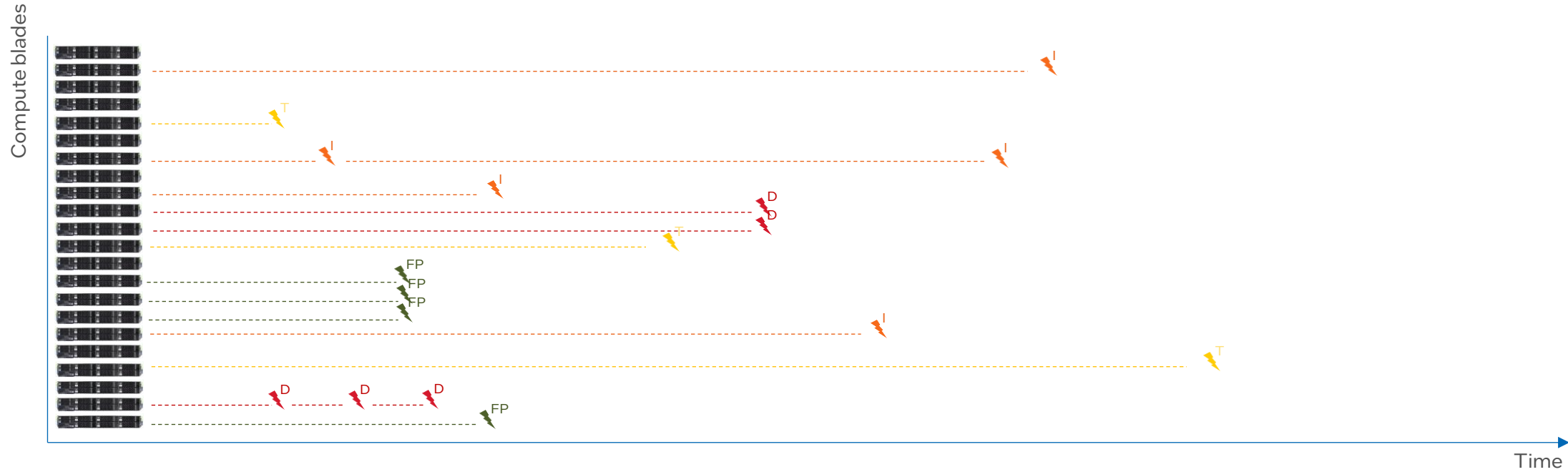
- Failures in leadership class accelerated HPC and AI systems have become the norm rather than the exception
 - This was anticipated a decade ago in the HPC community...
- As systems continue to scale in size, the frequency of failures on the entire system is expected to increase
- Tightly coupled parallel 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



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

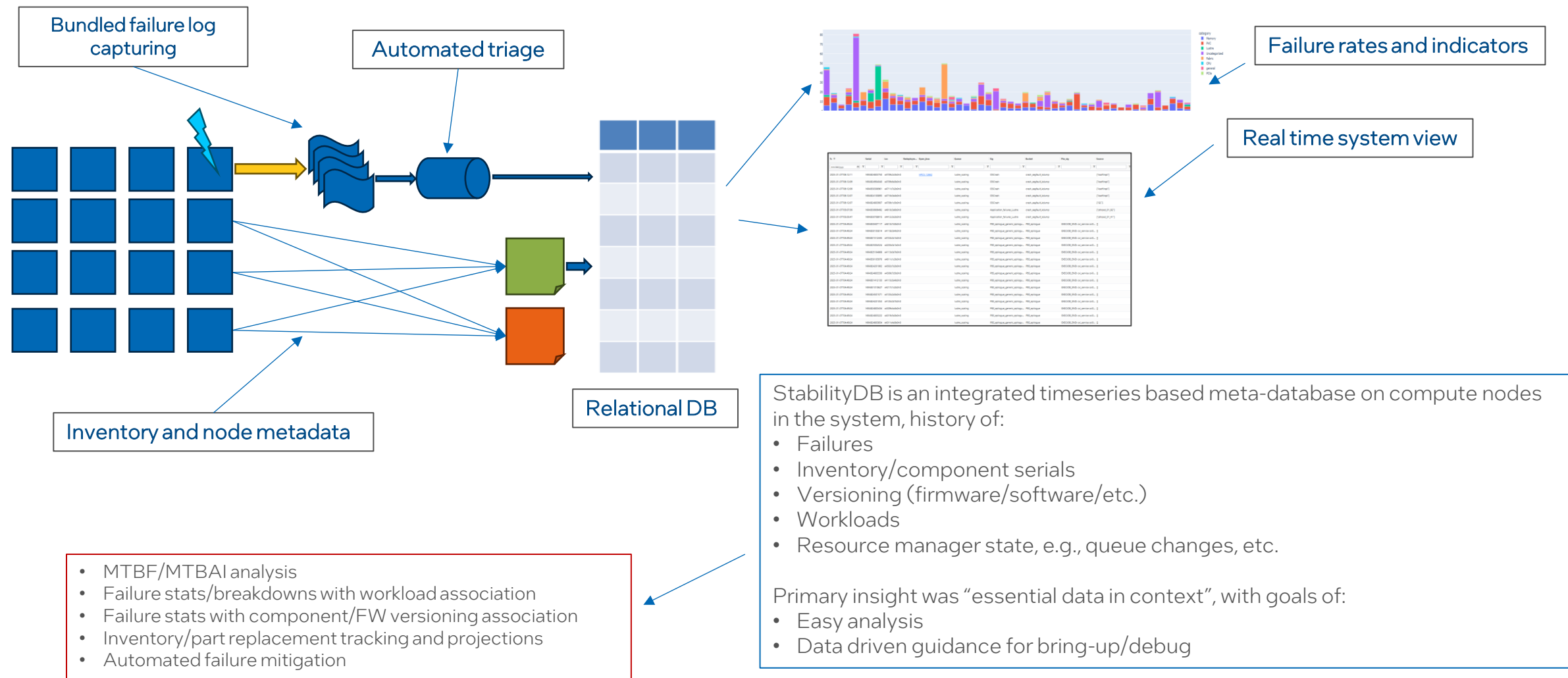
Key Observations and Problem Statement

- 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

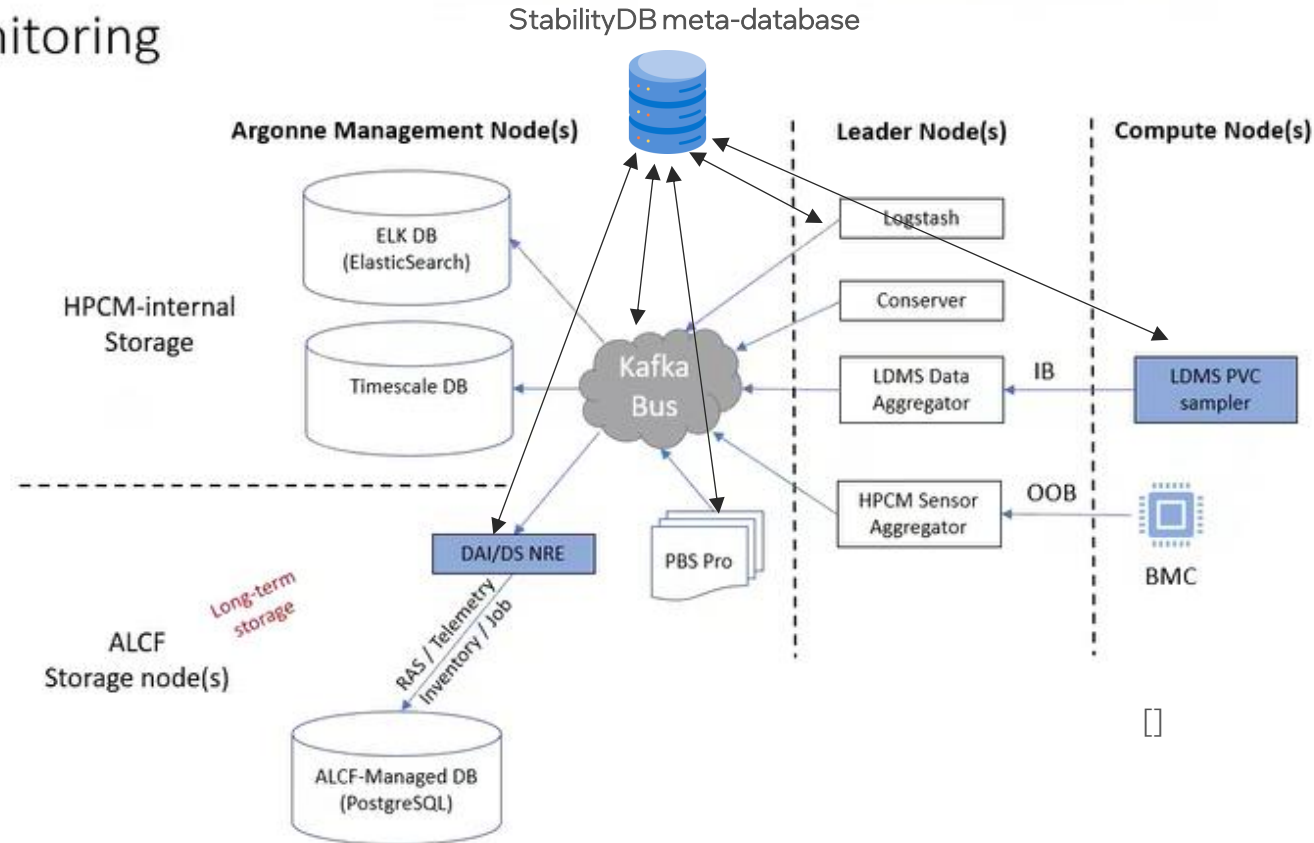
- Motivation
- Background
 - Aurora Overview
 - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

StabilityDB Infrastructure Architectural Overview



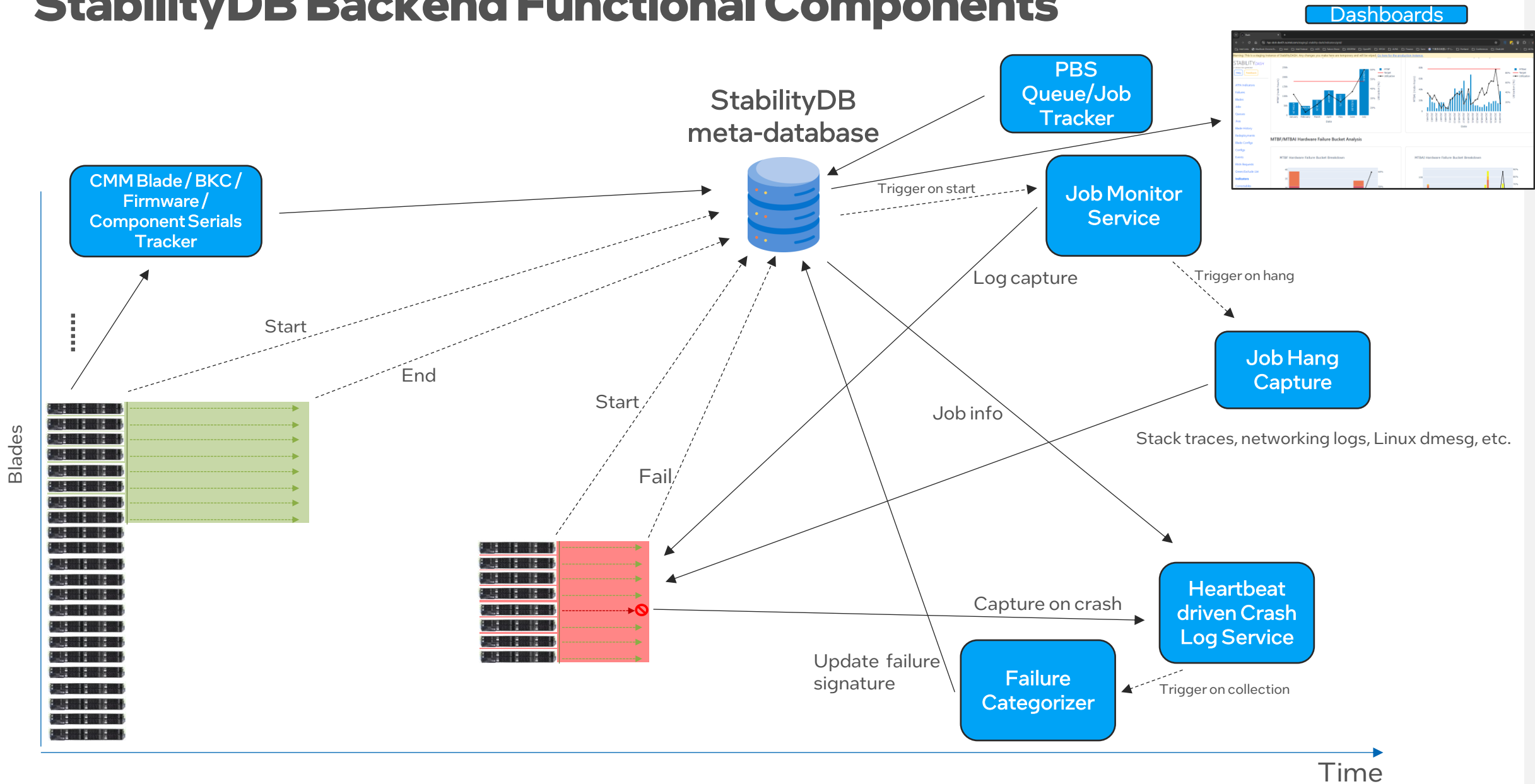
Aurora: Interaction with Cluster Management/Telemetry

Aurora Monitoring



- 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)

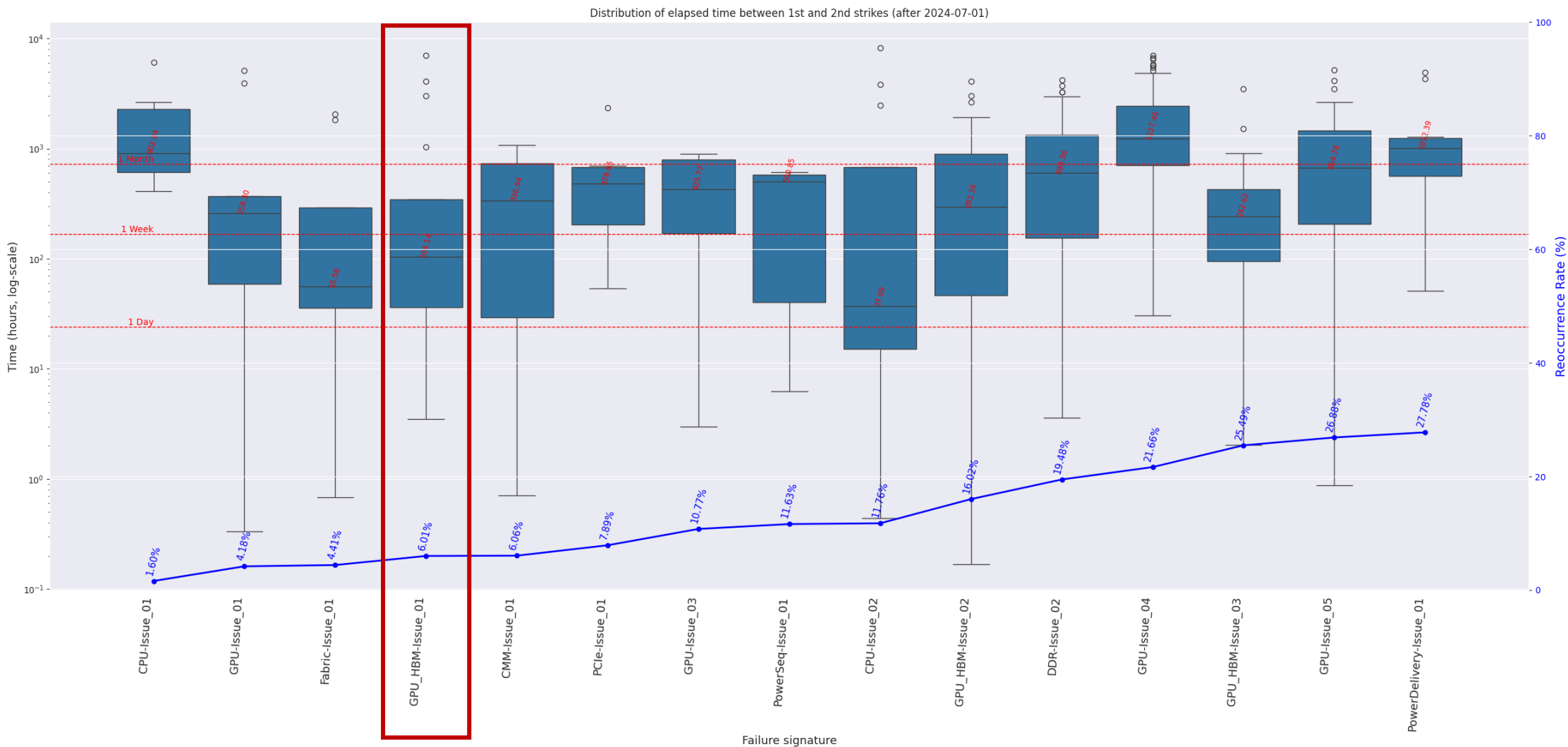
StabilityDB Backend Functional Components



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



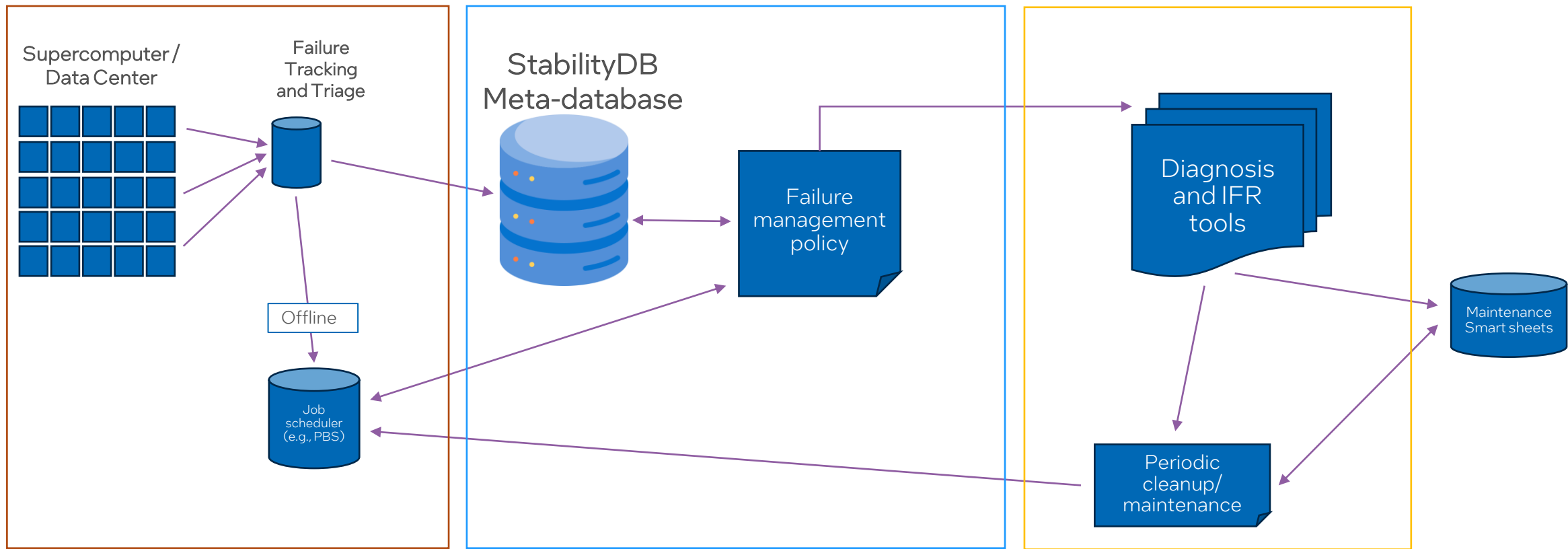
- Low reoccurrence rate
- Higher reoccurrence rate but long elapsed time between strikes

Fine-Grained Multi-Strike Policies

Updated_ts	Sig	Live	Criteria	Strike_1	Strike_2	Strike_3	Strike_4	Strike_5
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
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	fpv_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	h2_subslice_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



Failure triage and resource offlining

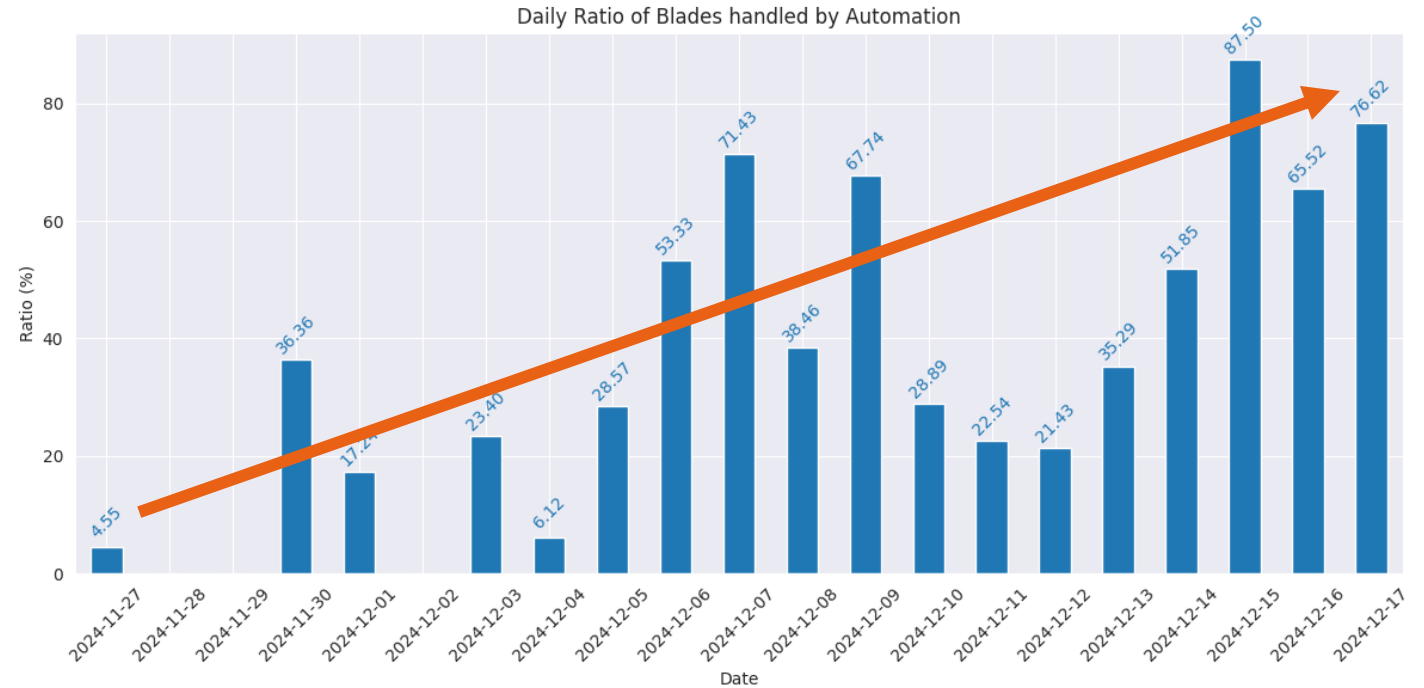
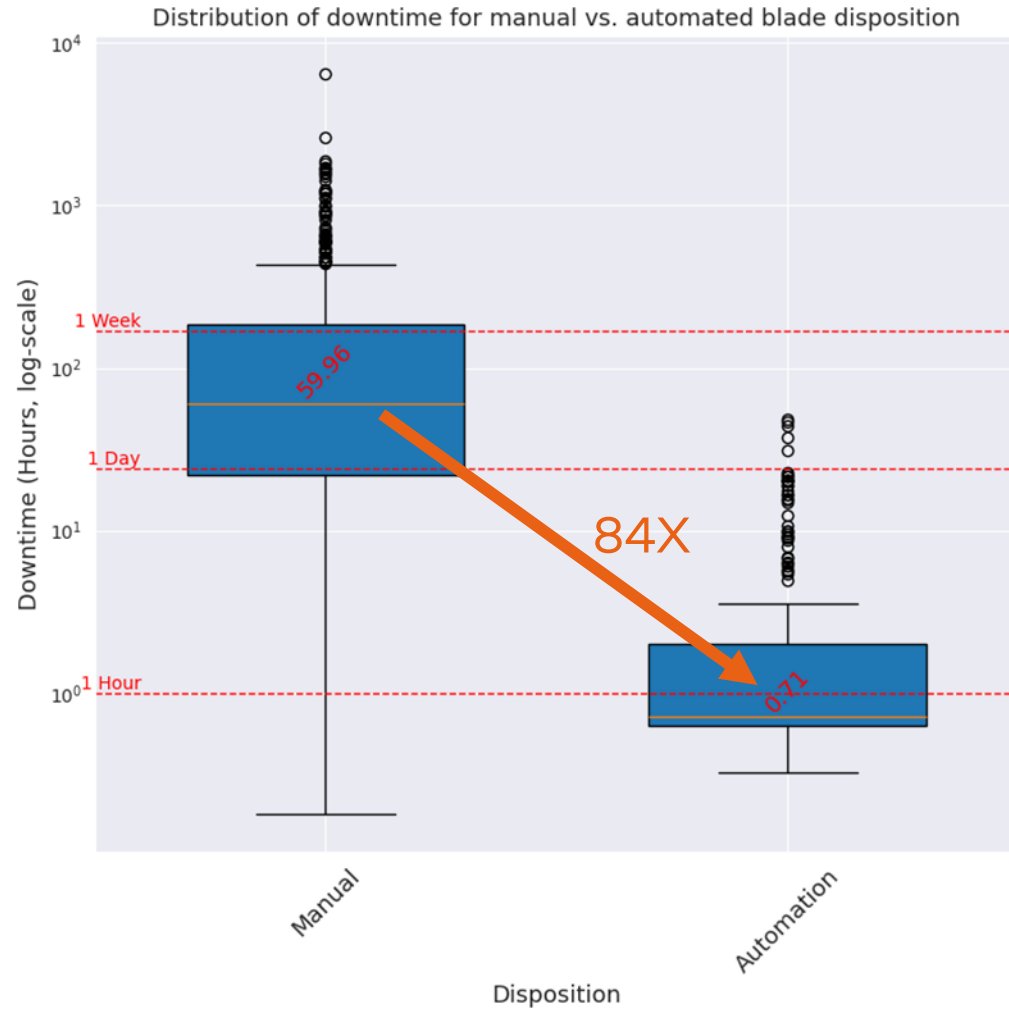
Historical data analysis and strike policy

Maintenance actions

Agenda

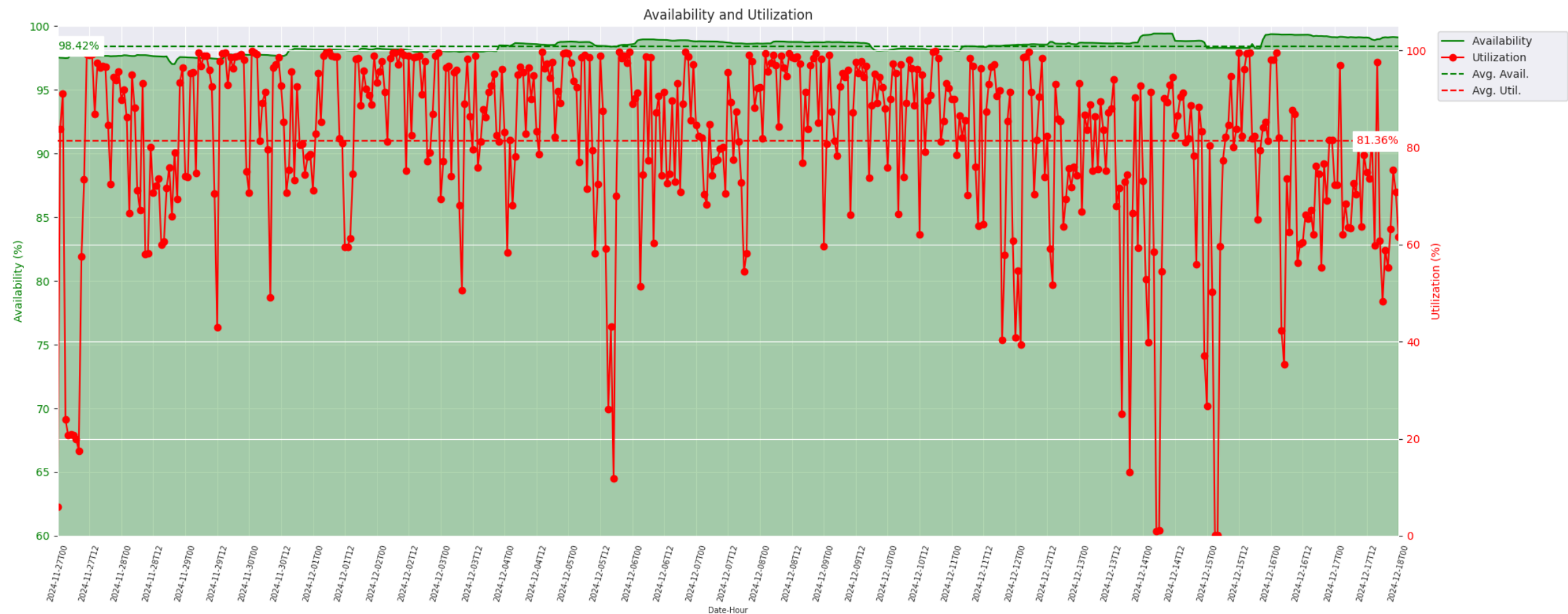
- Motivation
- Background
 - Aurora Overview
 - Failures in Large Scale Systems
- Problem Statement
- StabilityDB Architectural Overview
- Failure Strike Policy
- Failure Management Automation
- Results
- Conclusion

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

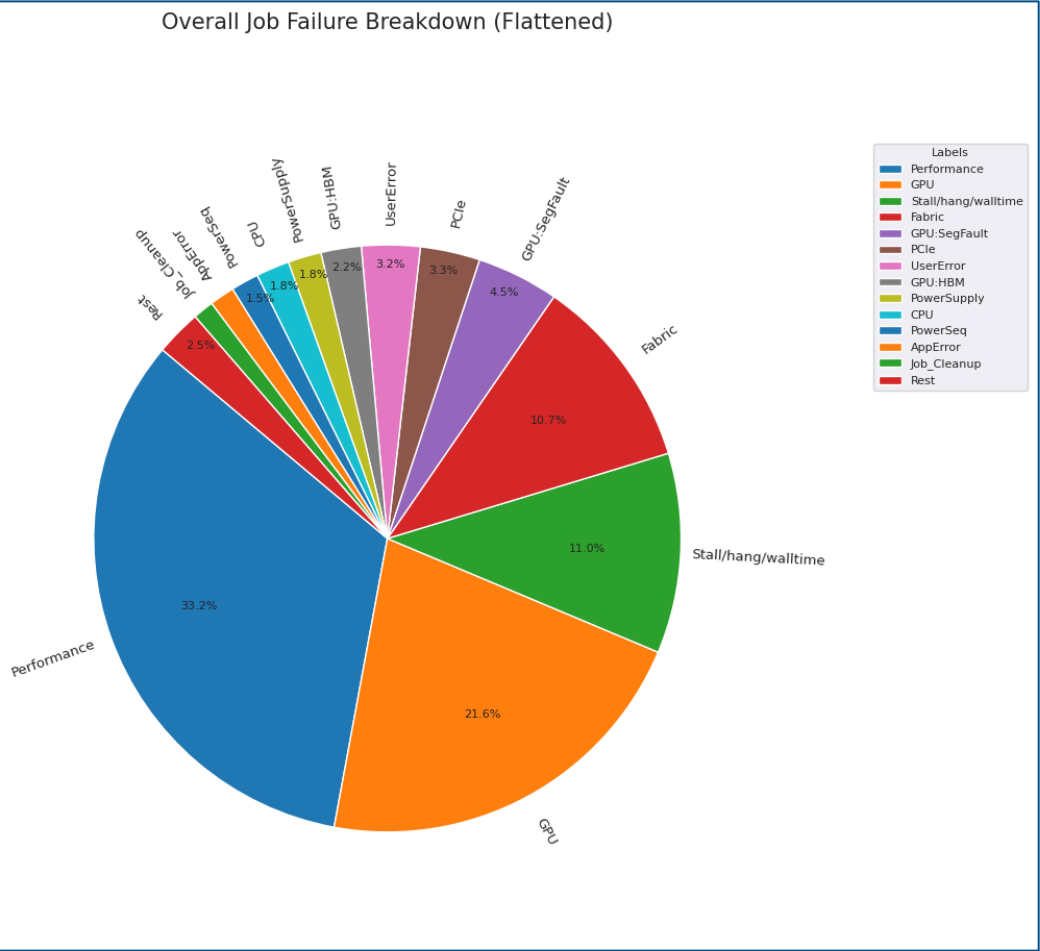
First and second strike counts per failure signature per month



- 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

Meta

The Llama 3 Herd of Models

Llama Team, AI@Meta
A detailed contribution list can be found in the appendix of this paper.

Modern artificial intelligence (AI) systems are powered by foundation models. This paper presents a new set of foundation models, called Llama 3. It is a set of language models that natively support multimodality, coding, reasoning, and tool usage. The largest model is a dense Transformer with 405B parameters and a context window of up to 128K tokens. This paper presents an extensive empirical evaluation of Llama 3. We find that Llama 3 delivers remarkable quality to leading language models such as GPT-4 on a plethora of tasks. We publicly release Llama 3, including pre-trained and post-trained versions of the 405B parameter language model and our Llama Guard 3 model for input and output safety. The paper also presents the results of experiments in which we integrate image, video, and speech capabilities into Llama 3 in a compartmentalized approach. We share this approach publicly with the state-of-the-art on image, video, and speech recognition tasks. The resulting models are not yet being broadly released as they are still under development.

Date: July 23, 2024
Website: <https://llama.meta.com/>

1 Introduction

Foundation models are general models of language, vision, speech, and/or other modalities that are designed to support a large variety of AI tasks. They form the basis of many modern AI systems.

The development of modern foundation models consists of two main stages: (1) a pre-training stage in which the model is trained at massive scale using straightforward tasks such as next word prediction or captioning and (2) a post-training stage in which the model is tuned to follow instructions, solve with human preferences, and improve specific capabilities (for example, coding and reasoning).

In this paper, we present a new set of foundation models for language, called Llama 3. The Llama 3 Herd of models entirely supports multimodality, coding, reasoning, and tool usage. Our largest model is dense Transformer with 405B parameters, generating information in a context window of up to 128K tokens. Each member of the herd is listed in Table 1. All the models presented in this paper are for the Llama 3.1 models, which we will refer to as Llama 3 throughout the paper.

We believe there are three key lessons in the development of high-quality foundation models: data, scale, and reasoning complexity. We seek to optimize for these three levers in our development pipeline.

- **Data:** Compared to prior versions of Llama ([Llama 2](#), [Llama 3.1](#)), we improved both the quantity and quality of the data we use for pre-training and post-training. These improvements include the development of new model pre-training and reasoning pipelines for pre-training data and the development of more specific quality assessment and filtering algorithms for post-training data. We pre-train Llama 3 on a corpus of about 15T multilingual tokens, compared to 1.5T tokens for Llama 2.
- **Scale:** We train a model at the largest scale than previous Llama models: our 405B language model was pre-trained using 3.6 x 10¹⁵ FLOPs, almost five times more than the largest version of Llama 2. Specifically, we pre-train a language model with 405B stronger parameters on 15.6T data tokens. As expected per

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.

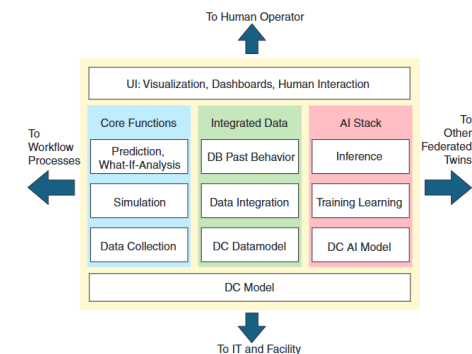
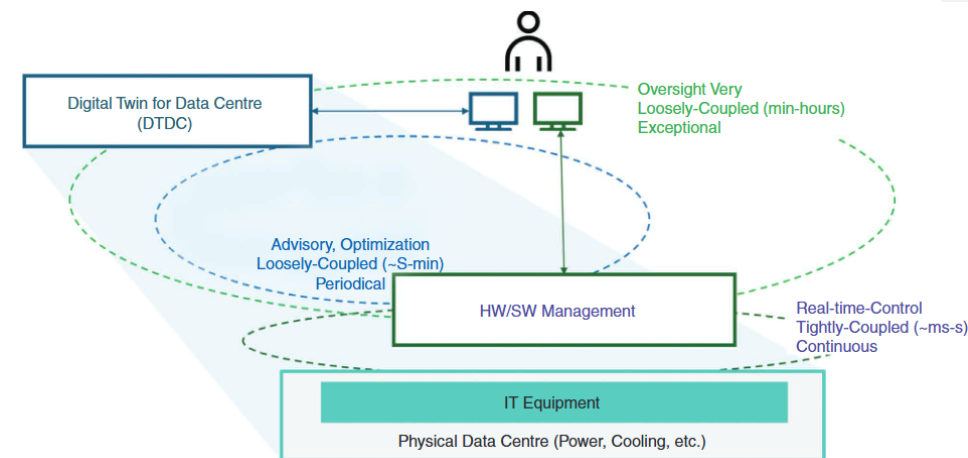


FIGURE 5. Data center digital twin software architecture. DB: database.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=10687340>



<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=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 Merrill
- Brian Holland
- Sucheta Raghunanda
- Ben Allen (ANL)
- Peter Upton (ANL)
- Doug Waldron (ANL)

Thank you for your attention!
Questions?

The Intel logo is centered on a solid blue background. It features the word "intel" in a white, lowercase, sans-serif font. A small, light blue square is positioned above the first vertical stroke of the letter 'i'. To the right of the word "intel" is a small white registered trademark symbol (®).

intel®