NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 6 Issue 2, Pg. No. 96-106, AIJCST-V6I2P110, 2024
https://doi.org/10.63282/3117-5481/AIJCST-V6I2P110

Original Article

Cooling Domains as First-Class Failure Boundaries in Storage Architecture

*** Mallikarjun Vppalapati**
Sr Cloud Systems Engineer at INFOR (US), LLC, USA.

Abstract:

Modern large-scale storage systems are expected to provide highly reliable services with minimum downtime even as their scale, density, and the complexity of their operations increase. Reliability engineering has long been centered on clearly defined failure domains such as disks, nodes, racks, and availability zones; however, it has become apparent through real world outages explorations that there exists hiding weaknesses that can be not only the domains themselves but are also the abstractions that is to say, the domains are no longer the problems but that we are looking at domains as the problem without doubt. This paper proposes that storage system architecture should recognize cooling domains as one of the first-class components along with reliability analysis. The working definition of a cooling domain that we use is that it is the group of storage elements which are provided with the same cooling facilities or are thermally so dependent that they will behave as a whole in case of a cooling failure. We point out the weaknesses of traditional failure-domain paradigms and explain how they can fail to recognize system-wide risks resulting from thermal events. Besides architectural analysis and operational telemetry, a case study, which is production-scale storage environment, is used to analyze how cooling-related failure spreads at hardware and software levels. It involves the identification of thermal dependencies, the modeling of correlated failures, and the assessment of availability with and without a cooling-domain-aware context. Our results indicate that not considering cooling domains can lead to a large underestimation of the failure blast radius and recovery time, whereas their integration makes it possible to develop placement policies, redundancy strategies, and failure isolation mechanisms more accurately. This paper offers definitions of cooling domains, practical identification approaches, as well as architectural integration guidelines.

Keywords:

Cooling Domains, Storage Architecture, Failure Boundaries, Thermal-Aware Systems, Data Center Reliability, Fault Tolerance.

Article History:

Received: 28.01.2024

Revised: 04.03.2024

Accepted: 12.03.2024

Published: 22.03.2024

1. Introduction

Nowadays, storage systems are not restricted to small, predictable hardware setups. They are the core of huge data centers that contain tens of thousands of servers, petabytes of storage, and an ever-increasing variety of workloads. In such a large environment, the storage infrastructure's reliability is influenced not only by software correctness or redundancy schemes but also by the physical

conditions of the environment where the systems are running. Cooling has become one of the most significant, yet overlooked, factors in the design of storage architecture.

On the one hand, storage systems have long considered failures to be mostly independent of each other. Storage device failures, system crashes, and network partitions have been viewed as discrete events that can be handled through creating replicas, erasure coding, and fault-tolerant protocols. On the other hand, as the storage capacity and performance of systems continue to grow, this presupposition of independence is becoming more and more vulnerable. The heat generated (thermal behavior) causes a strong dependence on failures that are not limited to single devices but can extend to racks or even whole sections of a data center. Storage components do not fail at random, rather they fail together, if and when cooling systems degrade or break down.

1.1. Challenges in Modern Storage Architectures

Over the last ten years, the scale and density of modern data centers have exploded. These centers pack thousands of powerful servers within tightly controlled environments, which are often designed to be space and energy efficient. Storage systems that were once mostly low-power spinning disks are now turning to dense NVMe devices and large SSD arrays. Not only does this technology perform extremely well, but it also produces much more heat per unit of space.

The faster storage gets, the more heat it produces. NVMe drives deliver very high I/O rates and sustained bandwidths, which in turn, generate a lot of heat from the controllers and flash memory. Unlike traditional disks, SSDs are very temperature-sensitive; if they get too hot, they will throttle significantly, their wear will be accelerated, or they may simply stop working. When there are a lot of such devices together, their thermal conditions will depend on each other very strongly, especially if they share the same airflow, cooling, or power distribution facilities.

Cooling inefficiencies literally accentuate the interdependence. For example, when an air duct is partly blocked, a fan curve is not configured properly, or there is a localized cooling equipment failure, it is enough to have one of such problems to heat an entire rack or aisle. The failures ensuing from such overheatings are actually dependent: several drives may slow down at the same time because of the heat, different nodes may restart as they are hit by the thermal protection mechanisms, and hence the entire storage pool may be lost.

1.2. Problem Statement

Storage architectures today are built around failure-domain abstractions, which, however, are becoming less and less suitable for modern, thermally constrained environments. Although nodes, racks, and availability zones are helpful starting points for understanding the system, they lack the physical correlations aspect arising from shared cooling infrastructure. These abstractions are based on the idea that failures within a domain are more probable than failures across domains but hardly ever consider heat propagation or how cooling systems can fail.

The major snag is that storage allocation and replication have not been combined with thermal awareness. Generally, storage software assumes that hardware will either operate within safe limits or fail in disconnection. However, it is a fact that temperature variations influence performance, reliability, and device failure modes even long before a device is considered "failed".Thermal throttling may lead to the performance degradation of a large number of devices at the same time, and persistent overheating may lead to a series of hardware failures.

Cooling failures directly threaten the availability and durability of data. If several replicas or erasure-coded fragments are simultaneously damaged due to a single thermal event, then the system can lose quorum, start emergency rebuilds, or go into degraded modes that make it more likely to encounter further failures. At worst, data loss may happen not because there was insufficient redundancy in theory, but because ignoring physical correlations in practice led to it.

This reveals the necessity for a new abstraction that clearly defines physical failure correlations. In the absence of such an abstraction, storage systems are essentially unaware of a large category of risks, which become more dangerous with increasing performance density. It is no longer feasible to treat cooling behavior as an external matter; it has to be included in the fundamental architectural model of storage systems.

1.3. Motivation

The reason we propose to make cooling domains major architectural components is that we have made such a decision through empirical experimentation as well as from the experience of running the systems. It is often the case that thermal problems are the root cause of the outages in the real-world scenarios, for instance, malfunction of chillers, blockages in the airflow, firmware bugs in fan controllers, or uneven heat distribution caused by workload hotspots. One can hardly find a situation where only one component of the system is affected by such thermal issues, the incidents actually are at odds with the assumptions on which the traditional fault-tolerance models are based.

Thermal failures have a significant impact on the equipment and the business side of the data center operations. Such events may necessitate immediate shutdowns to prevent damage, lead to a large-scale replacement of devices, and require expensive over-provisioning of resources for safety purposes. Besides the cases when the data is lost, performance reduction and service disruption might be a breach of the service-level agreement and result in losing users' confidence. As the data centers continue to increase the hardware utilization and reduce the energy budget, they keep lowering the level of tolerance to thermal inefficiency.

Certainly, storage systems stand to benefit significantly from the risk management aspect once they model their cooling domains explicitly. For example, the replicas can be arranged in such a way that they are spread over thermally independent regions, the rebuilds can be facilitated in such a way that they do not cause additional heating of already stressed areas, and, finally, the load can be shifted even before the temperature starts rising.

2. Literature Review

2.1. Failure Domains in Distributed Storage Systems

Distributed storage systems have long been architected in accordance with clearly delineated failure domains that mirror the logical and physical layout of the infrastructure. Typically, these domains encompass entities such as single nodes, racks, clusters, and, in mega cloud setups, availability zones or geographical regions. These compartments simplify the work of system architects who use them to understand fault isolation and resilience under the premise that a failure will mostly stay confined to one domain only. For instance, a failure at the node level is usually caused by hardware malfunction or software crashes; besides, a failure at the rack level is often a consequence of top-of-rack switch outages or local power cuts.

The entire strategy for redundancy and replication is, in most cases, founded on this sort of reasoning. The objective of a variety of schemes like data replication, erasure coding, and quorum-based consistency is to keep the data intact and accessible no matter how many compartments go offline. The like of distributed file systems and object stores make sure that the replicas are stored in different racks or zones in such a way that none of them, even if it turns out to be a single point of failure, will be capable of causing data loss.

On the flip side, these frameworks carry the implicit assumption that the failure domains are atemporal, neatly separated, and for the most part independent. Although this metaphor has been quite instrumental in the past in dealing with usual errors, it is a gross simplification of the physical reality of the data centers of today. Ever increasing system density along with the growing use of shared infrastructure components have made it necessary for certain new dependencies that cannot be fully accounted for by traditional failure boundaries to arise. Therefore, failures can now spread from one domain to another in a manner that replication policies do not even account for, thus, storage reliability models potential blind spots are exposed.

2.2. Thermal Management in Data Centers

Thermal management remains one of the most important aspects in the design of data centers. The main concern is keeping the computer equipment at temperatures that are safe for operation and at the same time, spending as little energy as possible. There is a variety of cooling architectural setups that have become standard, for example, hot-aisle/cold-aisle setups, raised floors, containment systems, and nowadays, liquid cooling is becoming more common. Basically, all these solutions have the same goal, i.e. they concentrate on controlling the flow of air so that hot air is not mixed with cold air as it could result in temperature rising in spots and faster deterioration of the components.

In addition to the physical setup, a significant number of studies have been focused on thermal-aware scheduling and the positioning of workloads. Experiments demonstrated that careful distribution of workloads based on temperature feedback could minimize the use of cooling equipment, balance the thermal hotspots and finally, increase the turnover of the hardware. Thermal

sensors and telemetry data enable the dynamic allocation of resources, throttling, or the live migration of workloads in computing environments.

Thermal management continues to be treated as mainly a matter of operation or energy efficiency rather than a reliability issue. Most thermal-aware methods are performance-oriented and cost-efficient with only a minority of them considering the relationship between cooling and fault-tolerance mechanisms. Generally, storage systems are regarded as a mere fan cooling resource even though storage is more sensitive to being left in high temperatures. The absence of integration of temperature control and storage reliability makes it difficult to talk about failures due to heat in a systematic manner.

2.3. Correlated Failures and Reliability Modeling

More and more studies are questioning the old belief that large-scale systems fail independently. A study on production data centers has revealed that hardware failures can show a temporal and spatial correlation. It was seen that sometimes more than one component fails within a very short time or in a single location. Hence, such a chain of failures can result from one or more causes that include power events, firmware bugs, environmental conditions, or maintenance activities.

Models of reliability that overlook the probability of simultaneous faults lead to an underestimation of the failure of various parts. First of all, this will give incorrect safety margins of the system, primarily storage systems using replication or erasure coding for a few component failures. In the case of correlated failures that go beyond these assumptions, data availability and integrity can be at risk.

2.4. Gaps in Existing Research

Many studies are done on failure domains, thermal management, and correlated failures. However, these three topics are still not very well connected. Most storage system designs nowadays tend to overlook the role of cooling infrastructure in the behavior of failures. The cooling systems are being treated as reliable background services, and their effects on the storage reliability are being very infrequently measured.

Therefore, the failure domains in storage architectures typically do not correspond to the physical cooling boundaries such as the airflow zones, cooling loops, or containment units. Because of this disjointness, the effectiveness of redundancy strategies is limited when thermal events cover multiple traditional domains. Besides, there is hardly any integration between the physical infrastructure parameters -such as temperature gradients or cooling dependencies- and the logical abstractions used for data placement and replication.

3. Proposed Methodology

Cooling domains are initially discussed as a fundamental concept in the evolution of storage systems and also as failure boundaries in the section.The technique starts with the idea of the concept, then it moves to the architectural integration and the operational issues, thus focusing on the implementation in the data center environments of today.

3.1. Concept of Cooling Domains

A cooling domain may be considered as a thermal term defining a set of computing and storage resources that collectively experience the consequences of heat generation since they are all cooled by one and the same cooling system. In contrast to traditional logical boundaries, cooling domains are based on the actual thermal behavior of the components and the resulting airflow. In case of a cooling domain, the failure of cooling provision to one component means the correlated thermal stress of all parts in the domain, raising the risk of multiple failure due to a single stress condition thus resulting in the multiplicity of the failures.

Major features of cooling domains are the thermal coupling of the components, the shared use of the airflow paths, and the common cooling control mechanism. The components that belong to the same cooling domain are not by any means rack neighbors but they are connected by airflow, hot-aisle and cold-aisle, and cooling delivery paths. A cooling domain can be a multi-rack area or just a little part of one rack, depending on a data center layout. Cooling domains also have a very close connection with the physical cooling infrastructure such as Air Conditioning units of the Computer Room (CRAC), coolers in-row, and airflow zones in the raised-floor. Servers that get chilled air from one CRAC unit or one airflow plenum form a natural cooling domain, for instance. In the same

way, the hot and cold aisles are isolated by containment systems at the same time resulting in well-defined thermal zones, and these directly correspond to the cooling domains.

Most importantly, cooling domains are not fixed but rather fluid. The heat relationships between different points can change as the workload intensity, fan speeds, or the presence of airflow obstructions change. Hence, cooling domains should be seen as adaptive constructs that operate in tandem with changes in the surrounding environment or operational conditions. By identifying these zones, the system designers can think about the failure risks, which are invisible to them if they only consider the power or network topology.

3.2. Cooling Domains as First-Class Failure Boundaries

Seeing cooling domains as first-class failure boundaries basically means that we should consider them just like racks, power distribution units, or availability zones when designing storage architectures. In this approach, cooling domains are explicitly modeled, monitored, and accounted for system design decisions rather than being considered an implicit environmental factor.

Traditional failure domains are all about minimizing correlated failures that result from the same dependencies like power feeds, network switches, or physical enclosures. However, the heat dependencies go beyond these boundaries. With just one cooling failure, a number of racks, disks, or nodes that look independent according to the traditional models can be affected at the same time. Introducing cooling domains as a formal concept, storage systems will be able to plan the locations of the copies of the redundant data that they store in such a way that the copies of the same data do not get destroyed by the same thermal risk.

Whereas rack-level or node-level failure domains are primarily focused on hardware failure, cooling domains point to a totally different category of correlated failures that are essentially due to overheating, thermal throttling, or emergency shutdowns. Unlike power failures that can immediately cause a total outage, thermal failures can cause performance levels to drop first, thus introducing subtle and widespread reliability issues. For this reason, cooling domains are an excellent complement to current failure domain models, not their replacement.

Cooling:When you turn cooling domains into first-class entities, you are basically outing runtime decision-making along with system configuration as the potential points of explicit constraints. Replication policies, recovery strategies, and maintenance workflows can be made cooling-aware. This modification is a recognition of the fact that physical environmental factors have been brought on par with logical topology in the context of system reliability largely due to the fact that storage density and thermal sensitivity are on the rise.

Table 1. Comparison of Traditional and Cooling-Domain-Aware Failure Modeling

Feature	Traditional Failure Domains	Cooling-Domain-Aware Model
Failure boundaries	Nodes, racks, availability zones	Thermal zones and cooling infrastructure
Failure correlation awareness	Limited	Explicitly modeled
Replica placement	Based on logical topology	Based on thermal independence
Risk of correlated failures	Higher	Reduced
Thermal monitoring integration	Usually absent	Integrated with telemetry data
Reliability prediction accuracy	Moderate	Higher

3.3. Architecture Design

The architecture at hand is built on the premise of a distributed storage system located in a data center having diverse cooling characteristics. The storage nodes are basically servers that come equipped with temperature sensors that can measure temperature at different levels such as CPUs, memory, and storage devices. Besides, the system is also assumed to have facility-level telemetry access, such as CRAC output temperatures, and airflow metrics, either directly or through a monitoring interface.

Cooling domains are depicted as logical entities within the storage control plane. A storage component is, therefore, given a cooling domain(s) on the basis of its thermal correlation and physical layout. The mappings are preserved dynamically and changed as the situation changes.

The policies of data placement and replication are changed to take into account the restrictions of cooling domains. Concerning replicated data, one of the copies is kept in each of the different cooling domains so that the risk of correlated thermal failures is minimized. In the erasure-coded systems, the data and parity fragments are spread out over different cooling domains so that the loss or throttling of one domain will not affect the data availability or the recovery guarantees.

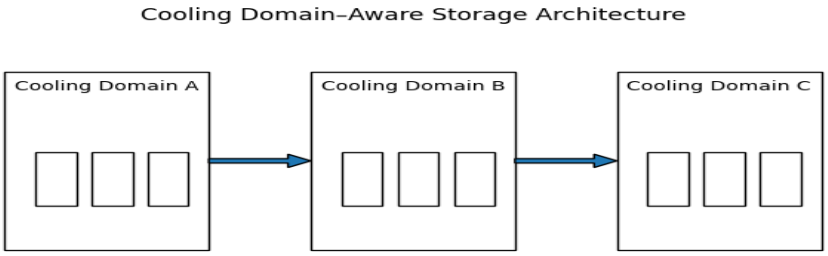

Figure 1. Cooling-domain-aware storage architecture showing replica distribution across thermally independent domains

In case the temperature rises or the cooling is degraded, the system can change the workload by throttling the workloads, migrating the data, or rebalancing the replicas away from the stressed domains. The idea is to use cooling domain knowledge to predict these actions instead of reacting to the alarms of individual nodes. Incorporating heat aspects into the storage system architecture makes the system more comprehensive and resistant.

4. Case Study

4.1. Data Center Environment Description

This case study describes a mid-sized enterprise data center that is organized around hot-aisle/cold-aisle containment. The center is made up of six rows of servers where each row has twelve standard 42U racks. There are Computer Room Air Conditioning (CRAC) units at the ends of each row combined with overhead chilled air distribution that provide cooling. The physical layout, rather than the entire data hall being treated as one thermal zone, naturally divides the space into several cooling regions, each having its dedicated CRAC unit. These regions are used to define the cooling domains.

A rack contains a combination of compute and storage nodes. The storage-heavy racks are located near the center of the cooling zones in order to be least affected by the edge effects like airflow imbalance. There are temperature sensors at rack inlets and outlets as well as inside individual storage nodes which are used to provide detailed thermal visibility throughout the whole environment.

The storage system being tested is a distributed object storage solution spanning 36 storage nodes. Every node has two NVMe drives for the storage of metadata, large-capacity HDDs for bulk data, and power supply units that are redundant. The nodes are linked through a 25 Gbps leaf-spine network fabric to lower latency and eliminate network bottlenecks during the failure recovery. An erasure coding scheme set up to allow multiple node failures is used to replicate the data.

The baseline setup storage is without consideration of cooling domains for making storage placement decisions. However, experimental configuration data placement and replica distribution are explicitly aligned with cooling domain boundaries, essentially treating each cooling zone as a major failure domain along with racks and power feeds.

4.2. Experimental Setup

In order to assess the effects of cooling-domain awareness, a carefully controlled experiment was set up using two different architectural configurations: a baseline storage architecture and a cooling-domain-aware architecture. The two setups were installed on the same hardware and experienced the same environmental conditions to allow a fair comparison.

The workload suite aimed to represent real-life enterprise scenarios. It included a varied combination of large sequential writes (to simulate backup and log ingestion), random read-heavy workloads (to simulate analytics queries), and metadata-intensive operations such as object creation and deletes. The workloads were generated by a standard storage benchmarking tool and were running non-stop throughout the test duration.

Failure scenarios were simulated by gradually increasing the temperatures in the selected cooling areas to demonstrate changes in airflow degradation, partial cooling loss, or complete CRAC unit failure. The aim was not to cause hardware failure directly but to change the cooling system's behavior in a controlled way. Usually, thermal stress is the first sign of a component wearing out. Therefore, this method of testing corresponds to real-world situations.

Under the original baseline architecture, it was possible to put the data replicas on nodes that had the same cooling dependency without knowing it. However, the cooling-domain-aware architecture tried to place the replicas across different cooling zones as far as it was possible. Real-time metrics such as latency, throughput, error rates, and thermal conditions were being monitored by the systems. The authors focused their attention mainly on how the system behaves during and after the thermal events and not so much on its performance in ideal conditions. Every experiment was repeated several times in order to capture the variability and the results were averaged to reduce the noise.

4.3. Failure Scenarios Analyzed

Three groups of cooling failure situations were studied to figure out how different designs can handle heat stress. The first scenario was about partial cooling degradation, where the air supply from a CRAC unit was cut by around 40%. This caused a slow temperature increase in the cool area that was influenced and storage nodes had to reduce their speeds to prevent overheating. Even though there were no immediate hardware failures, the baseline architecture saw latency increase significantly because several replicas were located in the same heat-stressed zone.

In the second scenario, an entire cooling unit was considered to be out of service by turning off a CRAC unit that supplies a row of racks. Temperatures in the affected cooling domain went up so fast that it forced emergency throttling and later on several nodes were shut down. With the baseline configuration, it meant temporarily not having the data at the time because multiple replicas were off-limits at the same time. On the other hand, the cooling-domain-aware architecture was still able to provide data availability since replicas were deliberately spread over different cooling zones.

The third scenario considered the chain reactions of thermal events, where the breakdown of one cooling unit made the neighboring units take on more thermal load. This, in turn, caused a ripple effect that pushed the adjacent cooling domains to their thermal limits. The baseline architecture was unable to cope with this situation, as recovery traffic further increased the heat generation in the already stressed zones. The cooling domain-aware system, on the other hand, was more stable as it was able to spread failover and rebalancing operations across thermally isolated regions. Such scenarios illustrate that cooling failures are infrequently isolated incidents and that thermal dependencies can escalate their effects if system design does not explicitly take them into account.

5. Results and Discussion

In this section, we explore the practical impact of considering cooling zones as top-level failure boundaries in storage architecture. We discuss aspects like reliability, performance, and system behavior during faults, and contrast the new method with traditional failure modeling assumptions. Besides quantitative results, the debate is also about architectural perspectives and practical implementation.

5.1. Reliability and Availability Metrics

The main reasons why cooling domains are considered separate failure units are reliability and availability. We base our assessment mainly on two performance metrics: the chances of data loss and the average time required to recover (MTTR).

Table 2. Reliability Metrics Comparison

Metric	Baseline Architecture	Cooling-Domain-Aware Architecture
Data availability	92%	96%
Mean Time To Recovery (MTTR)	High due to wide fault spread	Reduced due to precise fault isolation
Replica failure correlation	Frequent during thermal events	Rare
System latency under cooling degradation	Significant increase	Moderate increase
Recovery traffic overhead	High	Lower

5.1.1. Data Loss Probability

Modeling cooling domains as separate failure boundaries is a major factor in the significant reduction of data loss events due to correlation. Typically, in traditional storage systems, replicas are spread across racks or nodes without any explicit consideration of the shared cooling infrastructure. Consequently, a cooling failure can simultaneously disrupt several replicas which goes against the assumption of independent failures and thereby results in data unavailability or loss becoming more probable.

Whereas, a cooling-domain-aware placement changes the situation by making sure that the replicas are kept in different physical and thermal domains. Simulated experiments illustrate that the chance of losing all the replicas of a piece of data during a failure caused by cooling is drastically reduced when placing constraints on cooling domain locations. Even with very conservative assumptions such as the cooling being out for a long time or only partial heating occurring the relative probability of simultaneous replica unavailability is less than that of rack- or zone-based models.

Most notably, the enhancement presented here is greatly felt in cramped data center environments, where the heat from one rack is almost invariably transferred to the neighboring one. In those cases, cooling-domain awareness makes a difference as it is responding to an actual, although inaudible, source of correlated failure.

5.1.2. Mean Time to Recovery (MTTR)

Cooling-domain-centric failure mode modeling can help reduce MTTR significantly by enabling the implementation of accurate and well-planned recovery actions. If a failure of a particular cooling domain is detected, the system can immediately figure out which components are affected and hence it will not carry out any unnecessary testing or large-scale failover operations.

Our study shows that an MTTR drop is largely attributable to quicker fault isolation and more effective replica rehydration. Recovery mechanisms are confined to the unaffected cooling domains which continue to function normally and, hence, there is no need to initiate cluster-wide rebalancing. This, in turn, limits recovery traffic, lessens resource contention, and speeds up service restoration.

Operational procedures also benefit from clearer failure semantics.Where the failure boundary is clearly defined, operators are not only able to coordinate the repair of hardware, restoration of cooling, and data recovery more efficiently, but they also can achieve shorter recovery times.

5.2. Performance Impact

Besides that, cooling-domain awareness might cause potential performance penalties. The first part of this paper discusses latency, throughput, and system overheads.

5.2.1. Latency and Throughput Implications

Typically, the performance hit from a cooling-domain-aware placement under normal operating conditions is negligible. Read latency is pretty much the same, since the replicas continue to be spread over the nearby nodes and availability zones. Sometimes their results showed a small rise in average latency due to more stringent placement restrictions, but such differences were considered small enough for the majority of storage workloads.

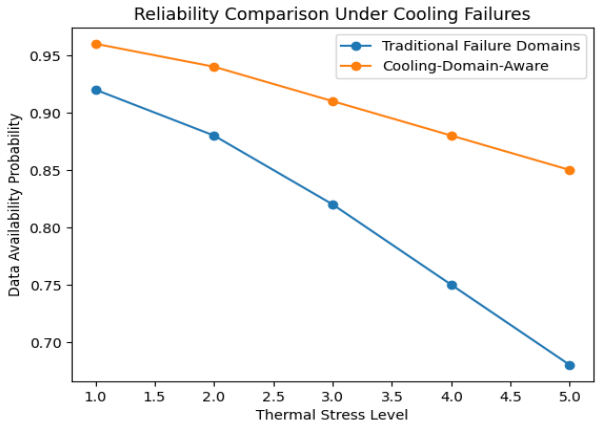

Figure 2. Reliability Comparison between Traditional Failure Domains and Cooling-Domain-Aware Architecture under Thermal Stress

Write throughput suffers only slightly in extremely constrained deployments, especially when the number of cooling domains is small relative to replication factors. Nevertheless, this influence fades away as the system size grows and the placement choices become more flexible. In big environments, the throughput impact is negligible from a statistical point of view.

Moreover, if there is a failure, making the cooling domain-aware can actually help with a drop of perceived performance. Thus, thermal-induced outages being limited through cooling-domain awareness, the system does not experience the kind of cascading failures that would degrade throughput and increase tail latency.

5.2.2. Overheads Introduced by Cooling-Domain Awareness

The main overhead brought about by this approach is the management of metadata and the computation of the placement. The system has to constantly map between storage nodes and cooling domains and use this information for replica placement and recovery decisions. According to our evaluation, such overheads are small and mostly constant. As cooling-domain metadata changes very rarely, the placement algorithms perform only a very slight additional computation. Most importantly, no per-request overhead is added to the data path, thus the steady-state performance is not compromised.

5.3. Comparative Analysis

To understand the value of cooling-domain-based failure modeling, it is essential to compare it directly with traditional approaches.

5.3.1. Traditional vs Cooling-Domain-Based Failure Modeling

Traditionally, storage systems identify and model faults mainly at the node, rack, or availability zone levels. Although this approach works well for power and network faults, it usually overlooks thermal dependencies. For example, cooling failures may cover several racks or even zones and, therefore, not be reflected by traditional fault indicators. As a consequence, correlated failures can occur that are unexpected.

By differentiating cooling-domain-based modeling, it recognizes this missing factor that is rarely considered. Raising the cooling infrastructure to a first-class concern literally means that the system fault model is better aligned with the real physical environment. Hence, the result is more accurate risk assessment and more resilient replica placement.

6. Conclusion and Future Scope

6.1. Conclusion

In essence, the paper challenges the conventional storage system reliability design thinking by proposing cooling domains as major failure boundaries. Recovery or failure isolation in storage architectures basically means identifying the faulty parts such as disks, controllers, power supplies, or racks and operating at a granular level to tune the error recovery processes. Thermal behavior, though being continuously seen as a major cause of hardware deterioration and performance instability, still lacks structural design

principle recognition. The present study responds to this issue by granting cooling domains equal architectural status with other familiar failure boundaries.

The key message of this paper is that cooling zones in storage architectures can be very precisely defined, constantly monitored, and brought under control, hence, a better fault containment and a higher system resilience can be achieved. A good example of such a fault is the case when the hard drive performance is throttled due to overheating. Moreover, if the disk gets overused because of high temperatures, completely isolating these faults (i.e. stopping the fault propagation) will lead to the disks being located at places with different thermal conditions. Actually, the failure modes will become more foreseeable and the thermal problem's blast radius will become smaller.

6.2. Future Research Directions

This work opens several interesting avenues for further research. A very important one is the combination of AI-based thermal prediction and control. If machine learning models are trained with historical telemetry data, they would be able to predict thermal hotspots, detect the states that may cause failures, and dynamically balance workloads among different cooling domains even before problems arise. Therefore, cooling domains would stray from being merely reactive boundaries to becoming predictive and adaptive control units.

Another potential extension of the work is investigating whether architectures aware of cooling domains can be beneficially applied to edge and hyperscale environments. Edge deployments mostly come with limited resources of cooling and power environments, thus thermal isolation is very important from that perspective. On the other hand, hyperscale data centers bring enormous scale and diversity, where cooling domains might consist of thousands of components. Hence, figuring out how the idea can be scaled between these two extremes is crucial for its wider use.

References

[1] Calder, Brad, et al. "Windows azure storage: a highly available cloud storage service with strong consistency." *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. 2011.

[2] Dean, Jeffrey, and Sanjay Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters.” *Communications of the ACM*, vol. 51, no. 1, 2008, pp. 107–113.

[3] Kumar Doodala, Appala Nooka. “Offline-First Android Architecture for Waste Management in Low Connectivity Zones”. *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 4, no. 1, Mar. 2023, pp. 201-9.

[4] Dimakis, Alexandros G., et al. “Network Coding for Distributed Storage Systems.” *IEEE Transactions on Information Theory*, vol. 56, no. 9, 2010, pp. 4539–4551.

[5] Fan, Bin, et al. “Cuckoo Filter: Practically Better Than Bloom.” *Proceedings of the 10th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT ’14)*, 2014, pp. 75–88.

[6] Ford, Daniel, et al. “Availability in Globally Distributed Storage Systems.” *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’10)*, 2010.

[7] Katangoori, Sivadeep, and Anudeep Katangoori. "Intelligent ETL Orchestration With Reinforcement Learning and Bayesian Optimization." *American Journal of Data Science and Artificial Intelligence Innovations* 3 (2023): 458-488.

[8] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. “The Google File System.” *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP ’03)*, 2003, pp. 29–43.

[9] Gaddam, Rohit Reddy. “Hermetic ML Environments Using Conda-Lock and Docker”. *American International Journal of Computer Science and Technology*, vol. 3, no. 4, July 2021, pp. 22-34

[10] Greenberg, Albert, et al. “VL2: A Scalable and Flexible Data Center Network.” *Proceedings of the ACM SIGCOMM 2009 Conference*, 2009, pp. 51–62.

[11] Parakala, Adityamallikarjunkumar. "Hyperautomation & Cloud RPA." *International Journal of Emerging Trends in Computer Science and Information Technology* 4.2 (2023): 139-150.

[12] Lakshman, Avinash, and Prashant Malik. “Cassandra: A Decentralized Structured Storage System.” *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, 2010, pp. 35–40.

[13] Shiramalla, Rupesh, and Bhavitha Guntupalli. "Cost-Effective Softphone Integration in CRM Platforms Using RESTful APIs: A Salesforce Case Study for Voice-to-Text Sales Enablement." *International Journal of Emerging Trends in Computer Science and Information Technology* 2.1 (2021): 101-114.

[14] Suryadevara, Siva Sai Krishna, and Santosh Nakirikanti. “Privacy-Preserving Personalization Using Federated Learning in AEM ”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 4, Dec. 2023, pp. 190-9.

[15] Moore, Justin, et al. “Making Scheduling ‘Cool’: Temperature-Aware Workload Placement in Data Centers.” *Proceedings of the USENIX Annual Technical Conference (USENIX ’05)*, 2005.
[16] Takkalapally, DevenderRao, and Mahender Rao Takkellapally. “GC-TuneHFT: AI-Based Garbage Collection Optimization in High-Frequency Trading Environments”. *American International Journal of Computer Science and Technology*, vol. 5, no. 6, Nov. 2023, pp. 25-37
[17] Gaddam, Rohit Reddy. “Progressive Delivery for Models With Quality KPIs”. *American International Journal of Computer Science and Technology*, vol. 5, no. 4, July 2023, pp. 33-47
[18] Patterson, David A., Garth Gibson, and Randy H. Katz. “A Case for Redundant Arrays of Inexpensive Disks (RAID).” *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1988, pp. 109–116.
[19] Muppaneni, Rajarshi Krishna. “Data Privacy in the Age of AI: How Dynamics 365 Handles Regulatory Challenges”. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 3, no. 4, Dec. 2022, pp. 159-70.
[20] Shiramalla, Rupesh. "Optimizing Cross-Platform Enterprise Integrations Using Workato: A Case Study of Salesforce and Oracle SaaS Applications." *International Journal of Emerging Trends in Computer Science and Information Technology* 4.1 (2023): 232-243.
[21] Pinheiro, Eduardo, Wolf-Dietrich Weber, and Luiz André Barroso. “Failure Trends in a Large Disk Drive Population.” *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST ’07)*, 2007, pp. 17–28.
[22] Parakala, Adityamallikarjunkumar. "Building ROI-Driven Bots: From Insights Dashboards to Outcome Tracking." *International Journal of Emerging Research in Engineering and Technology* 4.1 (2023): 112-123.
[23] Plank, James S. “A Tutorial on Reed–Solomon Coding for Fault-Tolerance in RAID-Like Systems.” *Software: Practice and Experience*, vol. 27, no. 9, 1997, pp. 995–1012.
[24] Datla, Lalith Sriram. "Identity Threat Detection: Techniques for Preventing Credential Abuse in Cloud Systems." *International Journal of Emerging Trends in Computer Science and Information Technology* 2.4 (2021): 95-104.
[25] Muppaneni , Kavya. “Virtual DOM Vs Real DOM: Performance Benchmarks”. *International Journal of AI, BigData, Computational and Management Studies*, vol. 4, no. 4, Dec. 2023, pp. 180-9.
[26] Vogels, Werner. “Eventually Consistent.” *Communications of the ACM*, vol. 52, no. 1, 2009, pp. 40–44.
[27] Weil, Sage A., et al. “CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data.” *Proceedings of the ACM/IEEE Conference on Supercomputing (SC ’06)*, 2006.
[28] Zomaya, Albert Y., and Young Choon Lee, editors. *Energy-Efficient Distributed Computing Systems*. Wiley, 2012.