

Matlab Simulations in Performance Analysis of Storage Area Networks

Jacek Nazdrowicz

Department of Microelectronics and Computer Sciences
Lodz University of Technology
Lodz, Poland
jnazdrowicz@dmc.s.pl

Maja Tuszynska

Cracow University of Technology
Cracow, Poland

EXTENDED ABSTRACT

Storage Area Networks (SANs) have become fundamental components of modern data infrastructures, supporting the rapid growth of cloud computing, embedded systems, and data-intensive applications. As demands for scalability, real-time responsiveness, and fault tolerance increase, SANs are tasked with providing consistent high-throughput and low-latency access to shared storage resources across heterogeneous environments. However, the complexity introduced by diverse hardware configurations, variable traffic patterns, and unpredictable fault scenarios poses significant challenges for ensuring reliable performance.

This paper presents a detailed simulation-based performance evaluation of SAN architectures, focusing on the interactions between system design, workload characteristics, and operational reliability. Using a custom-built discrete-event simulation framework implemented in MATLAB, the study models a SAN composed of three storage controllers with differing server capacities, service rates, and buffer sizes. The simulation captures realistic conditions by introducing bursty arrival processes, priority-based request scheduling, and stochastic controller failure and recovery events.

The workload is generated as a superposition of baseline Poisson arrivals and randomly occurring bursts, mimicking real-world fluctuations caused by backups, batch jobs, or synchronized tasks. Requests are categorized into three priority classes—high, medium, and low—with strict non-preemptive priority scheduling implemented at the controller level. Failures occur independently on each controller with a predefined probability, and recovery times follow an exponential distribution.

The simulation collects fine-grained metrics at 0.1-second intervals over a 1000-second window, tracking controller utilization, throughput, queue lengths, average waiting time, service time, latency, and dropped requests due to queue overflows. The results highlight several critical performance behaviors.

First, controller utilization and throughput show strong correlation, confirming that processing efficiency is directly tied to the load distribution mechanism. However, the use of a naive shortest-queue load balancing policy proves insufficient, leading to disproportionate load on faster controllers, particularly during burst conditions and failure-induced traffic redistribution. Controllers with lower service capacity become

saturated quickly, while more capable controllers are underutilized during certain periods due to delayed workload reallocation.

Second, queue length dynamics reveal system congestion patterns, with Controller 2 frequently reaching its queue limit. This leads to significant numbers of dropped requests, especially during failure periods when the remaining controllers are overloaded. Queue length spikes directly contribute to increases in waiting time and latency, emphasizing the importance of queue management in maintaining service quality.

Third, while the priority scheduling mechanism effectively ensures low latency for high-priority traffic—even during bursts—it results in unfair delays for medium- and low-priority requests. Starvation of lower-priority classes is particularly severe when bursts coincide with controller failures, revealing a trade-off between service differentiation and fairness.

The most alarming finding is the high volume of dropped requests—over 69% for each controller during the simulation period—primarily caused by queue overflows. This points to a lack of resilience in the current SAN configuration, underscoring the need for adaptive buffer sizing, dynamic admission control, and failure-aware load balancing.

The study concludes that while SANs are capable of delivering high performance under ideal or moderately variable conditions, they are vulnerable to instability under realistic workload and fault scenarios. Key recommendations include implementing smarter load balancing algorithms that incorporate priority, failure status, and controller capacity into routing decisions, as well as proactive fault mitigation strategies to redistribute workloads preemptively. Additionally, better workload forecasting and traffic shaping can help smooth demand spikes and reduce queue saturation.

This research provides valuable insights into SAN performance modeling and highlights crucial areas for architectural and algorithmic improvement. The simulation framework developed in this study can serve as a flexible testbed for further exploration of advanced resource allocation techniques, predictive failure models, and integration of emerging technologies such as NVMe-over-Fabrics or software-defined storage. Future work will extend this model to include network latencies, RAID-level behaviors, switch contention, and energy consumption, as well as investigate the use of reinforcement learning to enable self-optimizing SAN environments.