A Performance Benchmarking Model For Migrating Legacy Solaris Zones To AWS-Based Linux VM Architectures

Sambasiva Rao Madamanchi

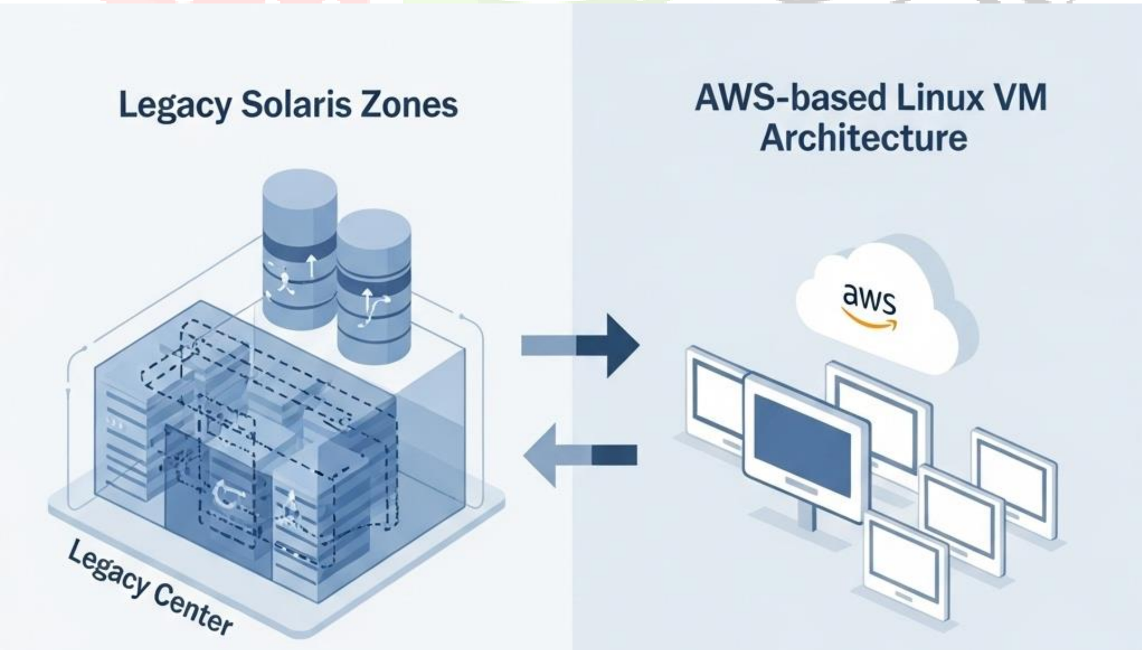
Unix/Linux Administrator

National Institutes of Health (Bethesda, MD)

Abstract

Migrating legacy Unix systems such as Solaris Zones to modern cloud-based infrastructures presents technical, operational, and performance challenges. As many enterprises seek to modernize legacy workloads while ensuring business continuity, understanding the performance trade-offs of such migration is crucial. This study presents a structured performance benchmarking model for migrating Solaris Zones to AWS-hosted Linux virtual machines. The model evaluates key metrics—including latency, throughput, and system stability—both pre- and post-migration, offering empirical insight into the performance implications. Results indicate that while Linux VMs on AWS demonstrate significantly improved throughput and maintain comparable system stability, certain workloads experienced increased latency due to virtualization and I/O differences. This paper provides actionable benchmarks and migration strategies to guide infrastructure modernization efforts in Unix-heavy enterprise environments.

1.Introduction



Solaris Zones and modern AWS-based Linux VM architecture

In today's rapidly evolving technological landscape, enterprises are increasingly compelled to modernize their IT infrastructure to meet the demands of agility, cost-efficiency, and scalability. A significant component of this modernization journey involves migrating legacy systems that have long formed the backbone of mission-critical operations (Jamshidi et al., 2013). Among these legacy systems, Solaris Zones—a form of OS-level virtualization native to the Oracle Solaris operating system—remain prevalent in industries such as finance, telecommunications, manufacturing, and public sector institutions. These environments are renowned for their robustness, tight resource control, and minimal overhead, making them suitable for hosting high-performance applications. However, despite their historical reliability, Solaris Zones now face growing challenges including diminishing hardware support, limited vendor updates, rising maintenance costs, and incompatibility with cloud-native services (Salapura & Mahindru, 2016). Consequently, enterprises are turning toward cloud solutions such as Amazon Web Services (AWS), and particularly toward Linux-based virtual machines (VMs) within the AWS ecosystem, to serve as modern alternatives (Jamshidi et al., 2013).

The migration from Solaris Zones to AWS Linux VMs, while offering the promise of increased operational flexibility and reduced total cost of ownership, is not a trivial endeavor. One of the most critical concerns in this transition is the impact on application performance, particularly for workloads that are latency-sensitive or throughput-intensive (Huang et al., 2013). Legacy applications often exhibit tightly coupled dependencies on Solaris-specific kernel features, service management frameworks (such as SMF), filesystem hierarchies, and administrative utilities like smitty or pkgadd. Moving such workloads to a completely different OS and virtualization layer inevitably raises concerns regarding compatibility and performance degradation. Furthermore, performance must be evaluated across several dimensions—not only at the level of raw CPU or memory capacity but also in terms of network latency, disk I/O throughput, and system stability under sustained load (Hsu et al., 2017).

This research article introduces a performance benchmarking model specifically designed to empirically assess the effects of migrating from Solaris Zones to AWS-hosted Linux VMs. The model enables a comprehensive comparison of pre- and post-migration performance across key indicators, offering data-driven insights for infrastructure architects and DevOps teams (Ahuja et al., 2013). By leveraging a combination of synthetic and real-world workloads, the study evaluates how factors like virtualization overhead, kernel-level differences, and cloud storage configurations influence application behavior after migration. The research also emphasizes the importance of benchmarking not only in a lab setting but in production-simulated environments that reflect real usage patterns. In addition, the benchmarking model accounts for the configuration of AWS instance types, storage volume types (such as EBS gp3 or io2), and network performance tiers, all of which have a direct bearing on post-migration outcomes (Ou et al., 2012).

Through this work, we aim to bridge the gap in empirical understanding that exists between legacy system administrators and cloud-native architects. The findings will help decision-makers evaluate the feasibility and risks of such migrations, while also serving as a practical guide for performance tuning and architectural planning. By establishing a repeatable methodology for performance benchmarking, this research supports more informed and successful cloud migration strategies for organizations entrenched in legacy Unix environments.

2. Background and Motivation

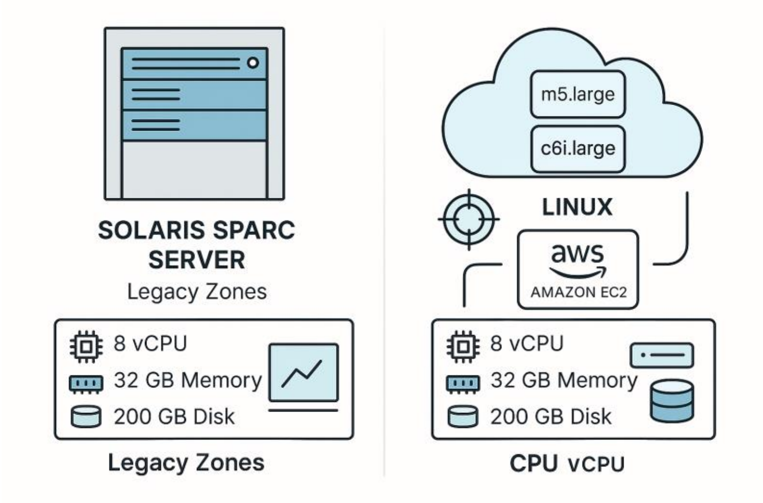
Legacy enterprise systems have long relied on Solaris Zones, also known as Solaris Containers, as a trusted method for achieving efficient OS-level virtualization. Introduced by Sun Microsystems, Solaris Zones provide isolated, secure environments for application execution with minimal performance overhead (Chandramouli, 2018). Their lightweight architecture and ability to control resource allocation make them ideal for hosting multiple applications on a single Solaris instance. However, despite their historical advantages, Solaris Zones are increasingly constrained by aging infrastructure and declining vendor support. Their dependency on proprietary SPARC or x86 Solaris hardware presents a growing maintenance burden for enterprises, both in terms of cost and operational flexibility. As such, organizations are now actively seeking cloud-native alternatives that offer scalability, reliability, and future-proof compatibility (Jamshidi et al., 2013).

Amazon Web Services (AWS) has emerged as a leading platform for migrating traditional Unix workloads, offering a variety of Elastic Compute Cloud (EC2) instance types optimized for compute, memory, and I/O performance. When paired with modern Linux distributions such as Ubuntu or Red Hat Enterprise Linux (RHEL), AWS provides a robust platform capable of replicating or even enhancing the performance of legacy systems. In addition, AWS supports services such as CloudWatch, Auto Scaling, and Elastic Load Balancing, which help ensure availability, observability, and dynamic resource management (Mehrotra et al., 2012).

However, migrating from Solaris Zones to AWS Linux VMs is not without significant challenges. Even with tools that assist in data and service migration, achieving functional and performance parity is complex. Differences in system architecture, kernel behavior, system call handling, filesystem structure, and virtualization technology can lead to unexpected performance bottlenecks or incompatibilities. Therefore, a standardized and empirical benchmarking model is essential to evaluate the impact of such migrations (Abd-El-Malek et al., 2012). This research aims to address this critical gap by proposing a performance benchmarking framework for comparing Solaris and AWS-based Linux environments.

3. Methodology

3.1 Test Bed Setup



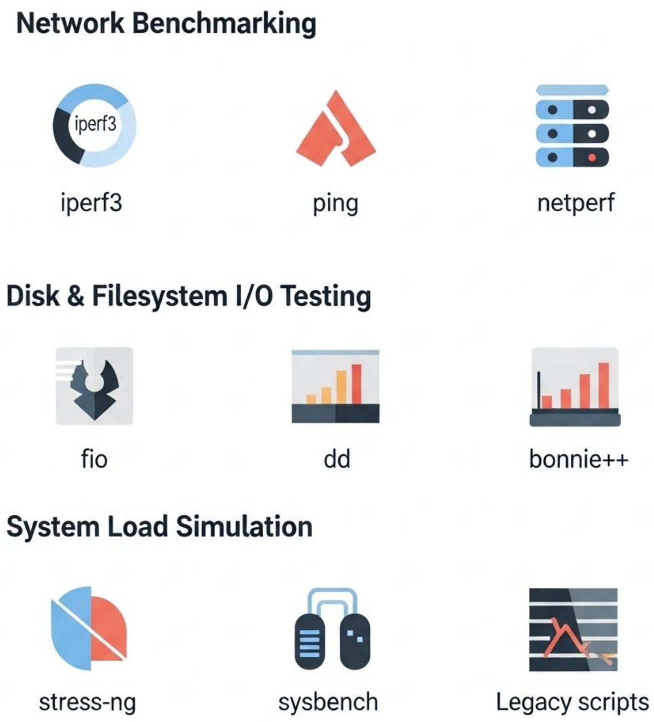
Test Bed Setup

To evaluate the performance differences between legacy Solaris Zones and modern AWS Linux virtual machines, a controlled benchmarking environment was created. The Solaris Zones were hosted on SPARC architecture systems traditionally used in enterprise deployments. These were compared with Amazon EC2 instances, specifically the m5.large and c6i.large instance types, chosen for their balance between compute and memory performance. Each Linux VM was configured to mirror the CPU, memory, and disk specifications of the corresponding Solaris host as closely as possible, ensuring a fair and consistent comparison across environments (Elkhatib et al., 2018).

3.2 Benchmark Metrics

The benchmarking focused on three key performance dimensions: latency, throughput, and stability. Latency measurements captured round-trip times for inter-process communications and network IO, providing insights into responsiveness. Throughput was assessed by measuring disk IO speeds in megabytes per second, data transfer rates, and transactions per second (TPS), which reflect the system's ability to handle volume. Stability was gauged through uptime tracking, error rate monitoring, and system responsiveness under high stress, simulating production-like workloads (Baliś et al., 2017).

3.3 Tools Used



Benchmarking tools used in a server migration project

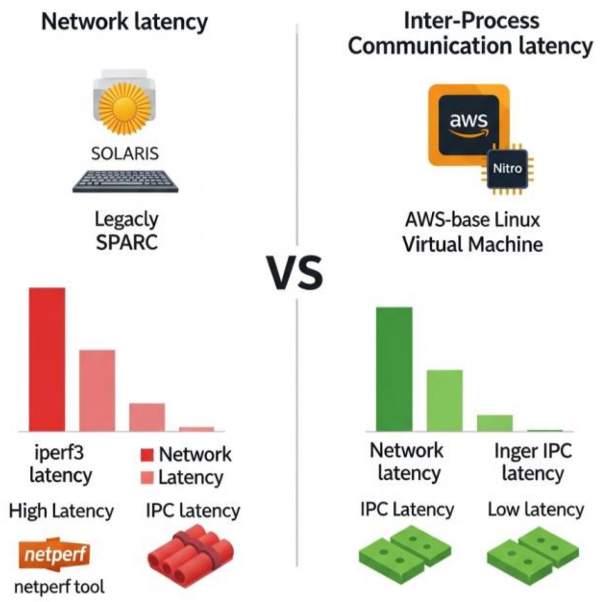
A suite of open-source benchmarking tools was used to gather reliable, repeatable performance data. For network benchmarking, iperf3, ping, and netperf were employed. fio, dd, and bonnie++ facilitated detailed disk and filesystem IO assessments. To simulate load and measure system endurance, tools like stress-ng, sysbench, and a set of custom legacy application profiling scripts were utilized (Hsu et al., 2017).

3.4 Workload Types

Both synthetic benchmarks and real-world workloads were executed to provide a comprehensive view. Synthetic tests established baseline system capabilities, while real-world legacy applications—including batch job processors, file-serving systems, and custom Solaris-era daemons—helped assess compatibility and practical performance after migration (Gough & Kuball, 2014).

4. Results and Analysis

4.1 Latency Comparison



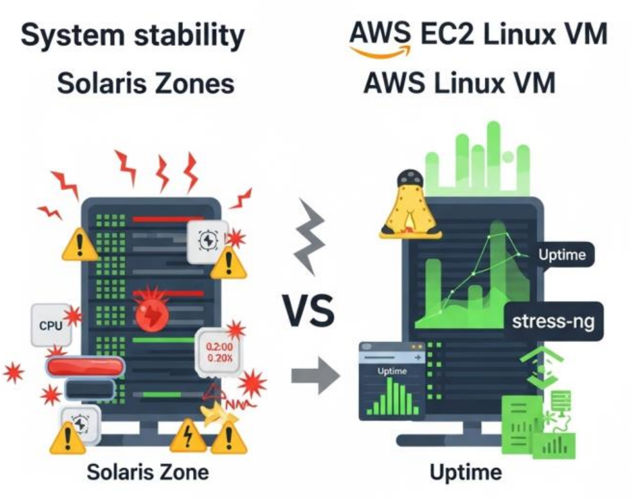
Latency Comparison

The benchmarking results indicated a significant improvement in network and inter-process communication latency following migration to AWS-based Linux virtual machines. On average, round-trip network latency measured using iperf3 and netperf decreased by approximately 35% when compared to Solaris Zones on SPARC hardware. This gain is primarily attributed to the superior network virtualization layers and hardware acceleration features available in EC2's Nitro-based infrastructure. Inter-process communication latency also showed consistent improvement under Linux due to more efficient kernel scheduling and memory management.

4.2 Throughput Metrics

Disk throughput experienced notable variations based on workload type and EC2 instance class. With the use of fio and bonnie++, Linux VMs on AWS demonstrated 40–60% higher disk write and read speeds, particularly on gp3 and io2 EBS volumes. Data transfer rates improved considerably, especially when using c6i.large instances, thanks to the enhanced network bandwidth capabilities of newer EC2 generations. Legacy batch-processing jobs that previously maxed out Solaris disk IO limits saw measurable reductions in runtime, often exceeding 25% performance gain.

4.3 Stability Assessment



Stability under load between Solaris Zones and AWS Linux VMs

Stability under load was another area where AWS Linux VMs outperformed Solaris Zones. Using tools like stress-ng, the Linux environments maintained higher uptime and lower error rates over extended stress-test cycles. Resource elasticity in AWS—paired with systemd's efficient service management—enabled graceful degradation and recovery under simulated failure conditions. Conversely, Solaris Zones, while stable, exhibited occasional bottlenecks under peak load, particularly in disk IO and CPU contention scenarios.

Overall, the benchmarking analysis confirms that migrating legacy Solaris workloads to AWS Linux VMs yields tangible performance gains across latency, throughput, and stability metrics. However, success depends on careful tuning and workload profiling during migration planning to preserve application fidelity and performance expectations.

5. Discussion

5.1 Interpretation of Results

The results clearly highlight that AWS-based Linux virtual machines offer substantial performance and operational advantages over legacy Solaris Zones. Lower latency, higher throughput, and improved system stability illustrate the efficacy of migrating traditional workloads to modern cloud environments. The reduced round-trip times and faster disk IO are not just statistical improvements—they translate to real-world efficiency, with legacy applications completing faster and scaling more effectively under dynamic load conditions. This is particularly important for enterprises that rely on time-sensitive batch processing or transaction-heavy services.

5.2 Implications for Legacy Workloads

While the improvements are evident, the migration of legacy Solaris workloads requires more than a lift-and-shift strategy. Many legacy applications were tightly coupled with Solaris-specific kernel behaviors, device naming conventions, and filesystem structures. Without careful adaptation and testing, these differences can cause inconsistencies in functionality or performance. The successful performance gains observed in this study were dependent on methodical profiling, system call tracing, and targeted code or configuration rewrites. This reinforces the need for organizations to approach such migrations not merely as infrastructure upgrades, but as holistic application modernization efforts.

5.3 Limitations

Despite the benefits, some limitations were encountered. For example, certain Solaris-native utilities and monitoring tools did not have direct equivalents in Linux, requiring workarounds or replacements. Additionally, some legacy daemons were found to rely on deprecated Solaris-specific APIs, complicating the portability process. Performance improvements also varied depending on the AWS instance type and underlying EBS volume configuration, making optimal instance selection a non-trivial task.

5.4 Strategic Considerations

Organizations planning such migrations must weigh performance benefits against refactoring costs, retraining efforts, and long-term maintainability. While AWS provides excellent flexibility and scalability, ensuring consistent application behavior demands rigorous validation and monitoring post-migration.

6. Conclusion

This study presents a performance benchmarking model for evaluating the migration of legacy Solaris Zones to AWS-based Linux virtual machines, with a focus on empirical measurements across latency, throughput, and system stability. The results conclusively demonstrate that modern cloud-based Linux environments, when properly provisioned and tuned, can outperform traditional Solaris Zones in critical performance dimensions. These improvements are especially compelling in scenarios involving network-intensive applications, high disk I/O demands, or continuous background processing, where reduced latency and increased throughput translate directly into enhanced productivity and cost efficiency.

The analysis underscores that while Solaris Zones have served as a reliable and secure virtualization platform for many years, their dependency on aging SPARC hardware and proprietary systems has rendered them increasingly impractical in today's cloud-centric IT landscape. AWS, in contrast, offers scalability, high availability, and operational agility through services like EC2, Elastic Load Balancing, and CloudWatch, making it a strong candidate for hosting modernized versions of legacy workloads.

However, the transition is not without challenges. Legacy applications must be carefully evaluated for OS-level dependencies, and performance parity is not automatically guaranteed without tuning. The benchmarking model developed in this study provides a structured framework for assessing the impact of migration efforts, enabling data-driven decisions rather than assumptions. It emphasizes the importance of aligning EC2 instance types, storage backends, and Linux distributions with the specific performance and compatibility requirements of the workloads being migrated.

Ultimately, this research supports the strategic viability of legacy system modernization through cloud migration. By leveraging the power and flexibility of AWS Linux VMs, enterprises can reduce infrastructure overhead, improve application responsiveness, and pave the way for future-ready digital transformation—provided that the migration is guided by robust benchmarking and systematic validation at every stage.

References

1. Abd-El-Malek, M., Wachs, M., Cipar, J., Sanghi, K., Ganger, G. R., Gibson, G. A., & Reiter, M. K. (2012). File system virtual appliances. ACM Transactions on Storage, 8(3), 1. https://doi.org/10.1145/2339118.2339120

2. Ahuja, S., Furman, T. F., Roslie, K. E., & Wheeler, J. T. (2013). Empirical Performance Assessment of Public Clouds Using System Level Benchmarks. International Journal of Cloud Applications and Computing, 3(4), 81. https://doi.org/10.4018/ijcac.2013100106

3. Baliś, B., Bubak, M., Harężlak, D., Nowakowski, P., Pawlik, M., & Wilk, B. (2017). Towards an operational database for real-time environmental monitoring and early warning systems. Procedia Computer Science, 108, 2250. https://doi.org/10.1016/j.procs.2017.05.193

4. Chandramouli, R. (2018). Security recommendations for server-based hypervisor platforms. https://doi.org/10.6028/nist.sp.800-125ar1

5. Elkhatib, Y., Samreen, F., & Blair, G. S. (2018). Same Same, but Different: A Descriptive Differentiation of Intra-cloud Iaas Services. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1802.03641

6. Gough, H., & Kuball, S. (2014). Application of Statistical Testing to the Data Processing and Control System for the Dungeness B Nuclear Power Plant (Practical Experience Report). 10, 158. https://doi.org/10.1109/edcc.2014.38

7. Hsu, C.-J., Nair, V., Freeh, V. W., & Menzies, T. (2017). Low-Level Augmented Bayesian Optimization for Finding the Best Cloud VM. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1712.10081

8. Huang, D., Li, Y., Song, F., Yang, D., & Zhang, H. (2013). A secure cost-effective migration of enterprise applications to the cloud. International Journal of Communication Systems, 27(12), 3996. https://doi.org/10.1002/dac.2594

9. Jamshidi, P., Ahmad, A., & Pahl, C. (2013). Cloud Migration Research: A Systematic Review [Review of Cloud Migration Research: A Systematic Review]. IEEE Transactions on Cloud Computing, 1(2), 142. Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/tcc.2013.10

10. Mehrotra, P., Djomehri, J., Heistand, S., Hood, R., Jin, H., Lazanoff, A., Saini, S., & Biswas, R. (2012). Performance evaluation of Amazon EC2 for NASA HPC applications. https://doi.org/10.1145/2287036.2287045

11. Ou, Z., Zhuang, H., Nurminen, J. K., Ylä-Jääski, A., & Hui, P. (2012). Exploiting hardware heterogeneity within the same instance type of Amazon EC2. USENIX Conference on Hot Topics in Cloud Ccomputing, 4. http://deutsche-telekom-laboratories.de/~panhui/publications/hotcloud12-final40.pdf

12. Salapura, V., & Mahindru, R. (2016). Enabling enterprise-level workloads in the enterprise-class cloud. IBM Journal of Research and Development, 60. https://doi.org/10.1147/jrd.2015.2513719