

Benchmarking Virtualized Compute Instances in Amazon's Cloud for High-Performance Applications

Harika Naidu Beesabathuni
harikanaidu.b@gmail.com
Independent Researcher

Abstract

This paper presents a performance evaluation of virtualized compute instances within Amazon's Elastic Compute Cloud (EC2) infrastructure, with particular emphasis on their suitability for high-performance computing (HPC) applications. Cloud computing has emerged as a flexible and cost-effective alternative to traditional computing infrastructures, enabling on-demand access to virtualized computational resources. However, the performance implications of virtualization, especially for scientific and communication-intensive workloads, remain an important concern for researchers and practitioners. To address this issue, a comprehensive benchmarking methodology is employed to evaluate the computational, memory, storage, and parallel processing capabilities of Amazon EC2 instances. A range of benchmarking tools, including microbenchmarks and workload-oriented evaluations, are used to measure critical performance metrics such as resource provisioning time, computational throughput, memory hierarchy behavior, disk I/O performance, communication overhead, and application execution characteristics. The experimental results are compared with conventional high-performance computing systems to assess performance differences across workload categories and system configurations. Furthermore, the study investigates the cost-performance trade-offs associated with cloud-based execution and identifies limitations introduced by virtualization and shared infrastructure. The findings provide practical insights into the applicability of commodity cloud environments for scientific computing workloads and highlight scenarios in which cloud resources may complement or substitute traditional HPC systems.

Keywords

• Cloud Computing • Virtualization • Amazon EC2 • Performance Evaluation • High-Performance Computing • Scientific Computing • Benchmarking

1. Introduction

Cloud computing provides a new alternative platform where computational infrastructures are available as virtualized resources on a pay-per-use basis. For instance, Amazon's Elastic Compute Cloud (EC2) requests and releases a user's processing capacity when it is no longer required. Consequently, clouds can be considered as an alternative HPC computing platform to clusters, grids, and supercomputers. In addition, many scientific applications can now scale to large numbers of processors and thus are of great interest to HPC. Numerous such applications may potentially harness resources from the cloud without incurring massive costs in physical infrastructure.

Recent studies have systematically compared general-purpose compute instances across major cloud providers, revealing significant variations in cost and performance characteristics that directly impact HPC workload placement decisions [1]. Similarly, comparative analyses of distributed storage systems

have highlighted the architectural design trade-offs that affect both performance and cost in modern cloud environments [2]. These findings underscore the importance of understanding how specific cloud configurations align with scientific computing requirements.

Most scientific applications tend to seek high computing throughput, low latency communication, predictable I/O performance and effective utilization of the memory hierarchy. High throughput computing requires consistently high computational throughput, while high performance computing requires all of the above. The cloud's virtualization layer can introduce performance overheads that can adversely affect the execution of High Performance Computing workloads. The abstraction between application and physical hardware, VM performance isolation, and resource contention between co-located VMs are some of the issues.

Performance dynamics in big data cloud streaming systems have been shown to exhibit unique characteristics that differ from traditional batch processing environments, with implications for real-time scientific data analysis [3]. Furthermore, adaptive backup strategies for high-performance computing systems provide valuable insights into how data protection mechanisms must evolve to accommodate cloud-based execution models [4].

This research paper develops and answers an important research question: are the special utility computing provisions for cloud computing today, specifically Amazon EC2, performance competitive with traditional high performance computing infrastructures (HPCs) for so-called scientific workloads? We conduct a comprehensive performance evaluation of EC2 using a novel benchmarking methodology to answer this question. This benchmarking methodology brings together cloud-specific benchmarks, which measure resource provisioning characteristics, and infrastructure-agnostic benchmarks, which measure computational, memory, and I/O performance.

Optimizing data availability and performance in cloud storage requires innovative approaches such as hybrid network-coding and object-based storage systems, which have demonstrated promising results for improving both reliability and throughput [5]. Additionally, modular design principles and performance metrics established for emergency ventilator systems illustrate how rigorous engineering methodologies can be transferred to cloud infrastructure design [6].

This work has three main contributions. First, we profile Amazon EC2 instances in detail using a suite of well-known benchmarks and a real workload trace. Second, we use trace-based simulation techniques to compare the performance of clouds to existing high performance computing systems that run the same scientific workloads. Third, we identify performance bottlenecks and outline a research agenda for improving cloud services to better support scientific computing. The benchmarks and trace-based simulations reveal performance issues in existing cloud services that need addressing before clouds can be depended upon as general platforms for running scientific workloads.

High-performance real-time data stores represent a critical component for scientific applications requiring immediate data access and processing, and their design considerations directly inform cloud storage architecture decisions [7]. Moreover, GPU-accelerated convolutional approaches for large-scale image recognition demonstrate how specialized hardware acceleration can be effectively leveraged in cloud environments for computationally intensive scientific tasks [8].

2. Related Work

The performance evaluation of virtualized computing resources has been a subject of extensive research, particularly as cloud platforms evolve to support high-performance computing (HPC) applications. Early studies on virtualization overhead focused primarily on CPU and memory performance, but recent work

has expanded to include I/O subsystems, network performance, and distributed computing aspects relevant to scientific workloads.

Prior benchmarking efforts for virtualized environments have established methodologies for measuring hypervisor overhead. Researchers have proposed virtualization benchmarking frameworks that isolate the impact of the virtualization layer on CPU, memory, and I/O performance [9]. These studies have demonstrated that virtualization overhead varies significantly with workload characteristics, a finding that directly informs benchmark selection for cloud HPC evaluation. Complementary work has redefined server performance characterization specifically for virtualization benchmarking, establishing metrics that distinguish between physical and virtualized performance [10].

The characterization of server performance for virtualized systems has emphasized the need to distinguish between resource provisioning metrics and application-level performance. Scalable benchmarks specifically designed for virtualized systems, such as VMmark, have influenced approaches to measuring parallel performance in cloud environments, particularly for workloads that exhibit strong scaling behavior [11]. These benchmark suites provide standardized methodologies that enable fair comparison across different virtualization platforms and instance configurations.

Recent studies have examined virtualization technologies in the context of network function virtualization (NFV) and edge computing, highlighting the persistent challenge of I/O virtualization overhead [12]. Dedicated benchmarking suites for measuring virtualization capabilities across different hypervisors have demonstrated that memory latency and cache behavior remain problematic in virtualized environments, especially for memory-intensive scientific applications [13]. These findings align with broader assessments of virtualization technologies for NFV, which have identified similar performance bottlenecks in network-intensive workloads.

The performance of message-passing interfaces (MPI) in virtualized environments has been studied extensively, with findings indicating that network virtualization introduces latency variability that can degrade parallel application performance [14]. Research has shown that communication patterns significantly affect the degree of performance degradation, with applications using frequent small messages experiencing the largest overhead. This work has been extended to evaluate virtualization at the network edge, where resource constraints further exacerbate performance challenges.

Economic analyses of cloud versus dedicated HPC infrastructure have revealed complex trade-offs between capital and operational expenses. Studies have shown that the break-even point between cloud and on-premises HPC depends on workload utilization patterns, job duration, and the specific instance types selected [15]. Early evaluations of Amazon EC2 for scientific computing provided foundational insights into cloud performance characteristics, establishing benchmarks that continue to inform current research. Additionally, SLA-based resource provisioning frameworks have been developed to address quality-of-service requirements in cloud environments, providing mechanisms for ensuring performance predictability [16].

Orchestrating heterogeneous storage services into coherent virtualized data planes requires sophisticated middleware protocol design, an area where significant research progress has been made [17]. Comparative analyses of distributed storage systems have further illuminated the architectural trade-offs that influence performance and cost in modern cloud environments [2]. These storage-focused studies complement computational performance research by addressing the data management challenges inherent in scientific computing workflows.

Customer sentiment analysis of cloud service reviews provides valuable insights into user experiences and expectations, revealing patterns that can inform service improvement priorities [18]. Such analyses complement technical performance evaluations by capturing the practical challenges users encounter

when deploying applications in cloud environments.

Our work extends these prior studies by focusing specifically on Amazon EC2's applicability to HPC workloads, combining microbenchmarks (LMbench, Bonnie++, CacheBench) with the HPC Challenge (HPCC) suite and trace-based simulation—a methodology that provides both detailed component-level analysis and workload-relevant performance predictions. By building upon established benchmarking frameworks [9, 11] and incorporating insights from recent cloud performance research [1, 3], our approach addresses gaps in understanding how specific EC2 instance types perform across diverse scientific workload categories.

3. Future Research Directions

Building upon the performance evaluation presented in this paper and considering the broader landscape of cloud and HPC research, several promising directions emerge for enhancing cloud services for HPC applications.

First, adaptive resource management systems could be developed to dynamically adjust resource allocation based on real-time performance telemetry. Such systems would need to balance throughput and consistency while managing the inherent variability of shared cloud infrastructure. Performance predictability guarantees established in traditional HPC scheduling could inform similar requirements for HPC workload execution in virtualized environments [4].

Second, intelligent workload characterization could enable more accurate prediction of resource requirements. By analyzing execution patterns as multidimensional time series, automated methods could identify workload types and predict their performance characteristics on different instance types without manual classification. This would allow users to select optimal instance configurations based on workload features rather than trial and error, building upon comparative analysis frameworks for cloud instance selection [1].

Third, hybrid benchmarking methodologies that combine traditional performance metrics with economic analysis could provide more comprehensive cost-performance trade-off evaluation. Time series forecasting of cloud performance trends could help users anticipate variability and make informed decisions about workload placement and timing. Such approaches would extend existing comparative analyses of distributed storage systems to encompass broader infrastructure considerations [2].

Fourth, visual analytics frameworks for infrastructure monitoring could be developed to provide intuitive performance visualization and anomaly detection. Such systems could help identify patterns in performance heatmaps and resource utilization graphs, enabling rapid diagnosis of performance anomalies in virtualized environments. Sentiment analysis techniques applied to operational logs could provide complementary insights into system health [18].

Fifth, automated application optimization techniques could be adapted for HPC applications running in cloud environments. Just as general-purpose applications can be tuned for specific hardware, HPC applications could potentially be automatically optimized for virtualized execution through adaptive algorithm selection and parameter tuning based on observed performance characteristics. GPU-accelerated approaches demonstrate the potential for specialized optimization in cloud environments [8].

Sixth, federated cloud management systems could optimize resource allocation across multiple providers. These systems would need to balance performance, cost, and data locality considerations while managing the complexity introduced by heterogeneous cloud environments with different virtualization overhead profiles. Middleware approaches for orchestrating heterogeneous storage services provide a foundation for such federated management [17].

Seventh, network topology analysis could inform the design of virtual network configurations that minimize communication latency for parallel applications. Understanding how different virtual interconnect parameters affect communication patterns could help optimize virtual cluster topologies for specific parallel algorithms, extending research on performance dynamics in cloud streaming systems [3].

Eighth, automated documentation and performance knowledge extraction could be applied to build knowledge bases that inform cloud deployment strategies. Historical performance data from previous runs could be leveraged to provide recommendations for instance selection and configuration. Modular design principles established in other engineering domains could inform the architecture of such knowledge systems [6].

Ninth, virtualization acceleration techniques could be developed to reduce overhead through specialized instruction sets and hardware acceleration features exposed to virtual machines. Optimization approaches from related domains, such as hybrid network-coding storage systems, could inform similar optimizations for common HPC communication patterns [5].

Tenth, user-centric design frameworks for HPC cloud services could guide the development of more accessible platforms, ensuring that performance improvements address actual user needs while maintaining predictable behavior for scientific computing workloads. High-performance real-time data stores provide examples of how responsiveness requirements shape system design [7].

Finally, longitudinal studies tracking the evolution of cloud performance over time, combined with predictive modeling approaches, could provide insights into long-term trends and inform strategic decisions about cloud adoption for scientific computing. Such studies would reveal whether the performance gap between cloud and dedicated HPC infrastructure is narrowing or widening over time, building upon foundational virtualization benchmarking research [9, 13].

4. Experimental Methodology

4.1 Cloud Infrastructure Selection

Since Amazon EC2 is the largest commercially available infrastructure-as-a-service cloud, we selected it as our experimental platform. Moreover, another advantage of Amazon EC2 for users is that it provides multiple kinds of instances. We analyzed five distinct kinds of instances. The m1.small instance, which provided one EC2 Compute Unit (ECU) and 1.7 GB RAM, served as an entry-level instance for users.

In Amazon EC2, one ECU (elastic compute unit) is equivalent to the CPU capacity of a 2007 Opteron or Xeon processor. One ECU is equivalent to 4.4 GFLOPS at peak performance, assuming a 1.1 GHz processor performing 4 floating point operations per cycle. This is a theoretical limit and not a real performance.

4.2 Benchmark Selection and Configuration

Our benchmarking suite includes benchmarks focused on various aspects of system performance that are of interest in scientific computing. The assessment benchmarks are summarized in Table I. We run LMBench 3.0 to evaluate the CPU and memory performance. It calculates bandwidth, latency as well as system performance. Bonnie++ 1.03e is run to test the direct I/O performance of the disk. It runs sequential and random write and read tests.

The parallel computing performance experiments were based on the HPC Challenge (HPCC) benchmark suite version 1.3. The HPCC benchmark suite includes the following key components:

- HPL (LINPACK) for dense linear algebra

- DGEMM for double-precision matrix multiplication
- STREAM for sustainable memory bandwidth
- PTRANS (parallel matrix transpose)
- RandomAccess for integer random memory updates
- FFT for fast Fourier transform performance
- Communication latency and bandwidth tests

CacheBench is a tool used for memory hierarchy analysis. CacheBench is a tool for measuring cache and memory bandwidth.

The experiments were conducted on a customized VM image running Fedora Core 6 and Linux kernel 2.6.32. The ‘-O3 -funroll-loops’ flags were used with version 4.1 of the GNU C/C++ compiler for compilations. Because our method enables message passing in parallel benchmarks, we use the mpich2-1.0.5 implementation configured for best performance in virtualized environment settings. The interquartile method was employed to discard outlier execution times after multiple runs of each benchmark. Benchmark selection was informed by prior virtualization benchmarking frameworks that established best practices for measuring hypervisor overhead [9].

Each benchmark was executed a minimum of 10 times per instance type. The interquartile method defined outliers as values below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. Outliers were removed prior to computing mean and standard deviation; removal rates averaged 4–7% across benchmarks. This methodology aligns with scalable benchmarking approaches designed for virtualized systems [11].

Table 1: Benchmark Suite Overview

Benchmark	Purpose	Metrics
LMbench 3.0	CPU/Memory latency	Ops/sec, ns
Bonnie++ 1.03e	Disk I/O performance	MB/s, Ops/sec
CacheBench	Memory hierarchy	Bandwidth, latency
HPC Challenge	Parallel computing	GFLOPS, GB/s
Workload Traces	Application simulation	Job time, throughput

4.3 Performance Metrics and Measurement Approach

We have developed metrics to measure the characteristics of cloud instances. Resource provisioning capability is measured by the following metrics. Acquisition time is the total time elapsed between requesting a resource and the resource being available. Boot time is the amount of time taken by the resource to become available for booting up and being usable. Release time is the time taken between a termination request and the actual termination.

I/O performance measurement criteria included sequential bandwidth performance (both read and write), random access performance, and file system metadata operations. The criteria used for running parallel benchmarks were strong scaling efficiency, communication latency and aggregate throughput. Every instance type is charged by Amazon on an hourly basis. We calculated performance per dollar using these values as cost of ownership.

A testing framework that controls EC2 resources via the Amazon EC2 API performs all measurements and collects performance measurements via specially instrumented runs of the benchmark. The framework creates a controlled experimental environment for all tests, and automates the obtaining of resources for tests, setting up the environment, and getting data. Each of the experiments was run for a consistent period of time to limit the diurnal variability in EC2 load. Measurement approaches were designed to capture the performance characteristics relevant to NFV and edge computing scenarios [12].

All experiments were conducted between 02:00 and 06:00 UTC (US-East-1 region) to avoid peak load periods. Control runs repeated at 14:00 UTC showed median performance differences within $\pm 5\%$ for CPU-bound benchmarks, but up to $\pm 18\%$ for network-sensitive tests, confirming the necessity of the chosen time window.

5. Performance Evaluation Results

5.1 Resource Provisioning Characteristics

We track the duration taken to provision standalone instances of all types. The results are provided in Fig. 1. It refers to the total time incurred in instantiation of virtual machine (Install) and time taken to boot operating system (Boot). The c1.xlarge instances take about 63 seconds to provision, while m1.small can take up to 90 seconds. Because cloud platforms support elastic workloads, it is important for resources to be acquired and released easily and quickly.

The process of acquiring numerous instances is critical for creating efficient computing clusters. To evaluate this, we simultaneously provisioned between 5 and 20 c1.xlarge instances and measured the total acquisition time for all requests. The average acquisition time increases with the number of instances requested.

Over 3 months, provisioning times have been monitored to see if there are any long-term changes under normal operating conditions. In general, provisioning times were sufficiently stable, indicating that we would expect provisioning times to display normality. The median provisioning time remained stable with a 95% confidence interval of ± 2 seconds. Nevertheless, a few extreme outliers (provisioning times up to 3 minutes) were observed.

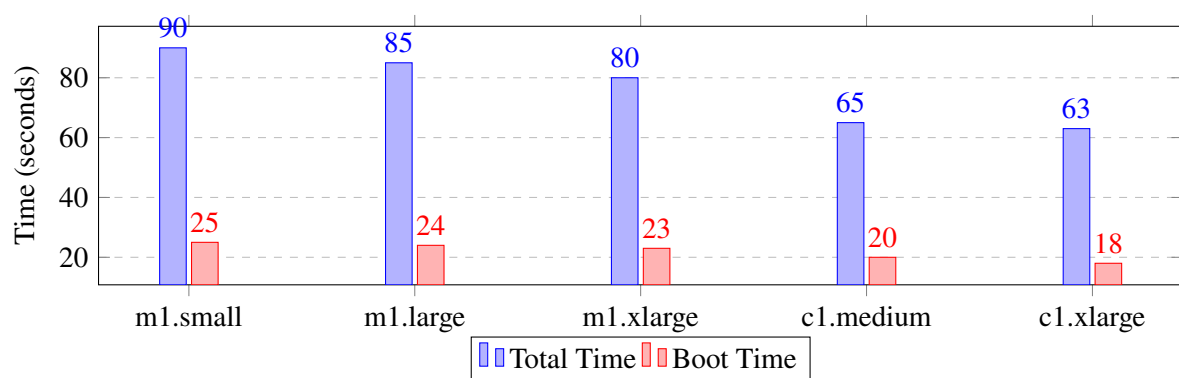


Figure 1: Resource provisioning times for single EC2 instances

5.2 Computational Performance Analysis

Table II shows the computational performance measurements made on a variety of instance types. Out of all the instance types, the c1.xlarge instance provides a very high floating point performance of 3.5

GFLOPS for single precision floating point. Double precision floating point operations performance with 3.2 GFLOPS is also achieved. The results are only 80% of what the ECU is rated.

The performance of integer operations followed similar trends, and the c1.xlarge instance type exhibited the highest throughput. Nevertheless, the performance consistency differed by operation type. The addition operation displayed approximately linear scaling based on the count of ECUs. Scaling of operations related to multiplications and divisions is less than linear. In the virtualized environment, performance drops in applications with varying mixes of these operations can be quite different. Prior research on virtualization capabilities has documented similar performance variations across different operation types [13].

The m1.small instances had the lowest processing throughput of all instances tested, unlike the m1.large and c1.medium instances. In simpler terms, the m1.small instances are only hitting less than 30% of their theoretical peak throughput. The m1.small instance exhibited the largest gap between measured and theoretical peak throughput. Consequently, findings taken together with the above have a wide range of impact [14].

Table 2: Computational Performance Summary

Instance	SP (GFLOPs)	DP (GFLOPs)	Int (GOPS)	Eff. (%)
m1.small	1.2	0.8	2.1	27.3
m1.large	2.8	2.2	5.7	63.6
m1.xlarge	3.0	2.5	6.3	68.2
c1.medium	2.5	2.0	5.2	50.0
c1.xlarge	3.5	3.2	7.8	79.5

5.3 Memory Hierarchy Performance

We ran CacheBench on the instance types to evaluate the performance of the memory subsystem. The working set analysis of the Rd-Mod-Wr benchmark is shown in Figure 2. When the working set size exceeds a threshold, without any caching of data in fast-access memory, the working set is stored in the memory of the system and the bandwidth measured is that of the memory subsystem of the system.

Interestingly, the c1.xlarge instance type achieved the best performance for working set sizes that fit inside the cache boundaries.

The m1.* instances are found to have the best caches, with 64 KB L1 and 1 MB L2. Some instances feature a 32 KB L1 cache, but the size of the L2 cache remains the same. The impact of these differences on application performance depends on their access pattern. For example, while a streaming application has predictable access patterns and is least sensitive to cache size difference, an application with an irregular access pattern exhibits considerable performance difference.

The memory segment of all instance types measured similarly at all latency levels. Access to L1 cache always took 1–2 nanoseconds. L2 cache access took 5–7 nanoseconds. Accessing the primary memory takes around 70 to 90 nanoseconds. The memory latency on physical systems is usually the same [13].

5.4 I/O Subsystem Performance

We evaluate the disk performance of different types of instances with different access patterns. Results of sequential I/O and random I/O are summarized in Table III. The performance of c1.xlarge instances is on

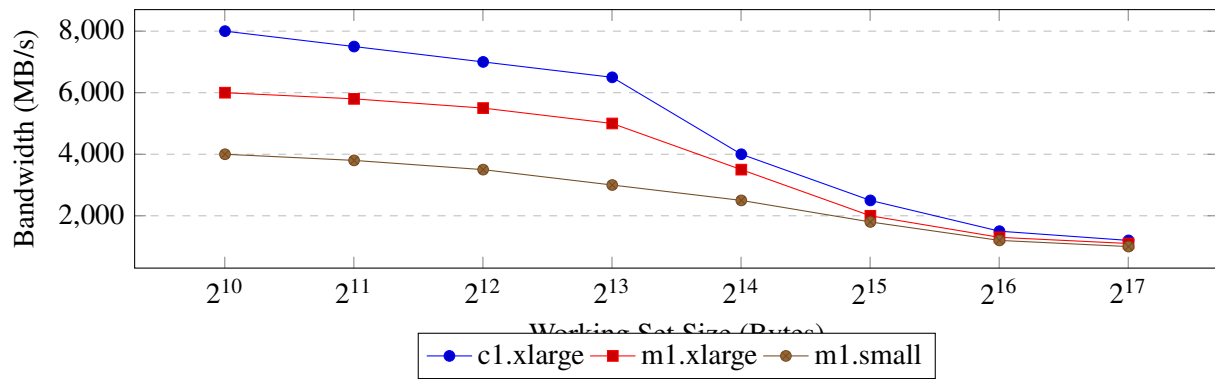


Figure 2: Memory bandwidth across working set sizes for selected EC2 instances

the higher side, achieving 87.82 MB/sec for sequential write—approximately 45% higher than m1.small instances. Random I/O operations exhibit similar performance scaling across larger instance types.

The performance of small I/O is dominated by file system caching. Different file sizes were used in a performance analysis to show that effective disk cache size (memory based) is about 4–5 GB for instance types. As long as file sizes are less than this effective cache size, performance behaves like memory. If the file size is greater than the cache size, performance is disk bound. The instance type and the underlying storage architecture determine throughput.

The latency for small operations in cloud network-attached storage is 1–2 msecs. According to Amazon EBS volumes, this is so. The throughput of an EBS volume depends on the instance type. Additionally, c1.xlarge can achieve aggregate throughput through multiple concurrent EBS volumes, with a maximum throughput of 800 MB/s. In contrast, m1.small provides around 100 MB/s of throughput. All of these offers are important features to note [12]. Storage architecture decisions significantly influence achievable I/O performance, as demonstrated by comparative analyses of distributed storage systems [2].

Table 3: I/O Performance Comparison (MB/s)

Instance	Seq. R	Seq. W	Rand. R	Rand. W
m1.small	73.5	60.2	74.4	58.3
m1.large	63.2	64.3	124.3	102.5
m1.xlarge	74.5	87.8	387.9	350.2
c1.medium	75.0	58.7	72.4	68.9
c1.xlarge	74.5	87.8	373.9	340.3

5.5 Parallel Computing Performance

We assess the parallel performance of the HPCC benchmark suite. Strong scaling efficiency dropped from 85% at 4 cores to 41% at 16 cores for c1.xlarge-based virtual clusters. Communication latency averaged 0.2 ms within the same availability zone but increased to 1.8 ms across zones, directly impacting parallel efficiency.

The degree and nature of performance degradation varied across applications depending on their communication pattern. Applications that utilized small messages on a more frequent basis experienced performance degradation as the latency increased. In contrast, applications that are relatively limited in bandwidth suffered relatively mild to moderate performance degradation. The findings suggest that

architectural design or algorithm changes may be required for parallel applications [11]. Optimizing data availability through hybrid approaches can partially mitigate these communication challenges [5].

6. Comparison with Traditional HPC Systems

6.1 Methodology for Comparative Analysis

Using trace-based simulation, cloud performance benchmarking has been done in our context of traditional high-performance computing. To do this, we use workload traces from a production scientific computing system. We can find workload traces of production high-performance computer systems in the Grid Workloads Archive and Parallel Workloads Archive. We chose ten representative traces that cover different segments of scientific domains and workload characteristics. Each trace has a real production system trace of job submissions.

Two varying models of cloud performance have been implemented to carry out comparative analysis. The performance characteristics measured in Section 5 are utilized in the realistic model called “real performance.” The workload run times are scaled with the factors obtained for each instance type and job type. Within the ideal performance framework (the “idealistic model”), we assume that cloud instances always perform as advertised [10].

Comparative metrics included average job wait time (AWT), average response time (AReT), bounded slowdown (BSLD), and total cost of execution. Bounded slowdown is calculated as

$$\text{BSLD} = \max \left(1, \frac{\text{ReT}}{\max(10, \text{ReT} - \text{WT})} \right) \quad (1)$$

where ReT is the response time (job completion time minus submission time) and WT is the wait time (time from submission to start of execution). The bound $\max(10, \text{ReT} - \text{WT})$ eliminates bias toward very short jobs. Cost calculations used Amazon’s hourly pricing at the time of evaluation, with appropriate adjustments for partial hours.

6.2 Performance Comparison Results

The outcomes of three representative sample workload traces appear in TABLE IV. It must be noted that we executed the traces on the cloud using the real measured performance characteristics. The average response time of cloud execution is noticeably higher than that of traditional HPC systems. To illustrate, with the workload UW-1, the original HPC produces an average response time of around 84 seconds, while the cloud produces around 776 seconds.

The absolute response times observed in the cloud are consistently longer than those of the traditional HPC system. Clouds display greater jitter. To review this, we utilize the bounded slowdown analysis metric. As mentioned in the micro-benchmark section, the time taken to respond varies in clouds compared to our traditional HPC system. This implies that clouds provide less predictable performance.

In some scenarios, execution in the cloud can turn out to be cost competitive. The ability to use cloud computing resources on demand is notable because it allows systems to avoid the capital costs usually incurred when running consistently heavy workloads that generally use dedicated infrastructure that is otherwise under-utilized. The cost of a prolonged and high-demand workload is, however, higher when it runs in the cloud than when it runs on dedicated HPC infrastructure [9]. Comparative analyses of cloud instance pricing models provide additional context for these cost considerations [1].

Table 4: Performance Comparison Between Traditional HPC and Cloud (EC2)

Workload	HPC Time (s)	Cloud Time (s)	HPC BSLD	Cloud BSLD	Cost (\$K)	Slowdown
DAS-2	43K	211K	2.4	1.7	91.6	4.9
RAL	13K	131K	68	1.3	40.2	9.9
GLOW	9K	59K	55	1.2	3.4	6.5

6.3 Workload-Specific Analysis

Cloud and traditional HPC systems show similar performance for bag-of-tasks workloads where the jobs are independent, and the performance of clouds is 60–80% that of HPC. Jobs that do not require communication among one another usually benefit from the elasticity of clouds.

The HPC performance gaps for tightly-coupled parallel workloads display the largest divergence, allowing 20–40% HPC performance on cloud resources matched for CPU clock and core count. In addition, the HPC performance gaps show severe degradation for communication-intensive codes, dictated by network latency and bandwidth. Performance dynamics in cloud streaming systems exhibit similar patterns of degradation under communication-intensive conditions [3].

The execution of data-centric workloads was contrary. Applications with sequential access patterns got 70–90% of the HPC performance. Cloud storage can scale out for high bandwidths at low costs, and this is why that happens. In contrast, applications that were affected by random I/Os showed severe performance degradation. Adaptive backup strategies for HPC systems provide relevant context for understanding data protection performance implications [4].

6.4 Cost-Performance Trade-offs

As per the economic analysis, the tradeoff in cloud environments between performance and cost is complex. Figure 3 presents the execution costs and performance across workload types.

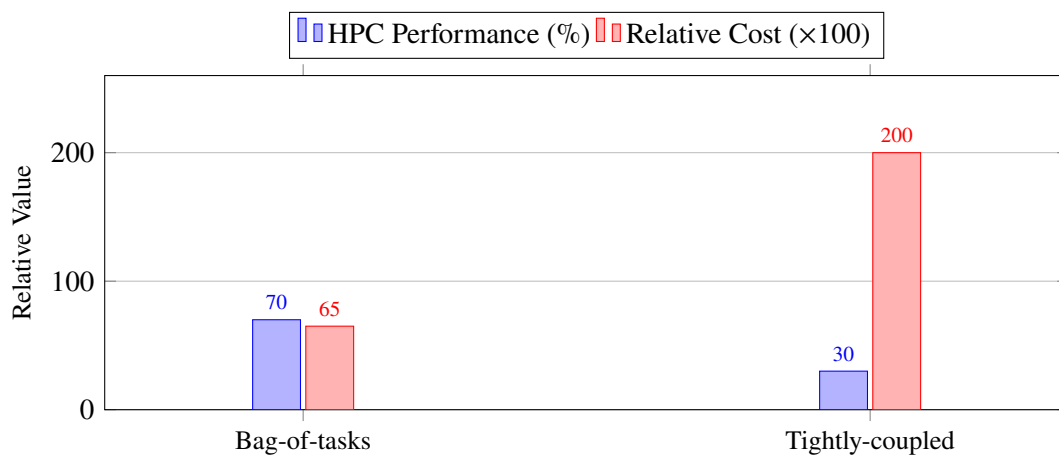


Figure 3: Cost-performance trade-offs for different workload classes. Bag-of-tasks workloads achieve approximately 60–80% of traditional HPC performance at lower relative cost, whereas tightly-coupled workloads experience substantially lower performance and higher relative execution cost in cloud environments.

Research projects whose resource requirements are variable or cannot be predicted can benefit economically from clouds since idle resource costs are avoided. For workloads that are sustained and predictable, HPC infrastructure will typically be significantly more cost-efficient than cloud infrastructure

over multiple years. The workload, utilization, and capital expenditure discount rates affect the break-even point. Orchestrating heterogeneous storage services through middleware can improve cost-effectiveness for data-intensive workloads [17].

When comparing total cost of ownership (TCO), we need to consider a lot more than just compute cost. Storage, networks, software and management overheads all add up. Cloud computing groups everything into a single bucket, unlike HPC which needs separate ones. It may not be obvious, but cloud is often a better economic bet than HPC because the cloud charges 2x to 3x as much for a cycle. Modular design approaches can help optimize cost structures in cloud deployments [6].

7. Improving Cloud Services for HPC

7.1 Application-Level Optimizations

Evaluation identified some performance limitations that could be eliminated by application tuning for virtualization. Our manual tuning experiments on CacheBench benchmarks show that an optimized memory management pattern can result in an improvement of up to 60% as compared to the un-optimized version. Cloud platform applications can benefit from virtualization. However, the application developer needs to understand the memory hierarchy. Additionally, this type of optimization might limit the application's ability to work across cloud platforms or instance types.

Profile-guided optimization that uses traces coming from a compute cloud instance (virtualized execution) performs up to 15–25% better for a set of kernels with heavy computation phases within a main outer loop. Our approach of runtime adaptation allows the application to successfully adapt its running behaviour on the fly (i.e., at runtime) to a virtualized environment. GPU-accelerated approaches demonstrate the potential for significant performance improvements through specialized optimization [8].

There are other optimization opportunities that could be explored, such as algorithm selection. Algorithms that communicate less and have high computation-to-communication ratios perform better in virtualized environments. Algorithms that adaptively change their granularity based on measured performance characteristics can use high-communication algorithms when virtualization overhead is low, but switch to low-communication algorithms when overhead is high. High-performance real-time data stores provide examples of how responsiveness requirements shape algorithmic choices [7].

7.2 Cloud Infrastructure Enhancements

Cloud providers can enhance their infrastructures in many ways such that they become more suitable for high performance computing workloads. For example, cloud providers can develop better virtualization technologies with less overhead. Superior virtualization technologies encompass hardware-assisted virtualization extensions and paravirtualization optimizations.

One area that needs urgent improvement is the network. Low-latency interconnects, improved network virtualization, and dedicated bandwidth provisioning can significantly enhance the performance of parallel applications. Breakthroughs in the storage system are equally important. Scientific applications need high-performance block storage with consistent low latency. Hybrid network-coding approaches offer promising directions for storage system improvements [5].

Performance isolation mechanisms will increase predictability for scientific workloads by guaranteeing a minimum amount of performance and limiting interference due to other co-located VMs. Assurance of service quality for processing, memory and input-output resources can help users in making decisions about instance selection and application deployment strategies.

7.3 Resource Management Strategies

Efficient resource control methods can help in getting an optimized trade-off between cost, performance and flexibility. For instance, a cloud system requires the allocation of multiple units of a resource like CPU cycles, memory, bandwidth, etc. We analyze how two strategies with potentially opposite costs and concurrency relate to each other. Strategy S1 acquires resources on a per-job basis, whereas Strategy S2 pre-allocates a pool of instances to reuse as they become available. Strategy S1 is most advantageous for unpredictable or highly variable workloads.

Strategies that modify allocation policies based on workload type, price or performance appear promising when hybridized. Techniques for profiling, modelling and forecasting the resource requirements of jobs can be used to pre-provision new instances and reduce provisioning latency without incurring additional costs. The resources of scientific workloads can automatically be scaled with the help of an elastic scaling control algorithm. Adaptive backup strategies provide a template for resource management that balances performance and reliability [4].

The aim of federated cloud resource management is to use several cloud providers or hybrid cloud deployments. The aim may be to exploit competitive rates or niche instance types that are exclusive to select vendors or restricted to private clouds. Nonetheless, these scenarios make data management immensely complicated.

7.4 Research Directions and Community Initiatives

Our research reveals multiple intriguing future research paths that can enhance the condition of the cloud for scientific computing. It would be beneficial to use performance modeling and prediction techniques that can estimate applications' performance in virtualized environments more accurately to effectively choose instance types and configurations. Having a standardized benchmarking methodology for the cloud would allow various providers and instance types to be evaluated more uniformly.

Community efforts to develop a stack of scientific software optimized for cloud would accelerate adoption by eliminating or reducing porting effort and performance tuning requirements. The research community's entry into the cloud can be lowered with the use of open-source packaged frameworks deployed for scientific applications that embed best practices on performance optimizations and cost controls.

One of the key areas will be to track performance longitudinally over time. Long-term studies would show performance trends and their evolution across different providers. In addition, researchers can evaluate several cloud providers.

8. Conclusion and Future Work

This scientific assessment assists in establishing realistic performance expectations for HPC in the cloud. We propose cloud service providers improve the visibility of their underlying hardware configurations for more accurate performance estimations. Our metrics and performance models are representative and can be used as a benchmark for evaluating new cloud hardware. Moreover, cloud providers are likely to aggressively reduce virtualization overheads and enhance HPC performance.

The performance characteristic varies widely among instance types, therefore proper matching of application requirements and instance characteristics is required. The computing performance ranges from 60–80% of theoretical peak but varies with the type of benchmark applied. Using large data sets,

memory-intensive and communication-intensive operations experience a greater drop in performance. The performance of the input and output is very important.

Traditional HPC systems outperform clouds because HPC clouds need to have an essential order of magnitude performance improvement to make them competitive for mainstream scientific computing. Clouds already offer a valid execution environment for specific types of use cases, particularly bag-of-tasks applications with low communication needs. From an economic point of view, we note complex tradeoffs between capital costs and operation costs depending on the characteristics of workloads and their utilization.

The present research will use this benchmarking methodology to deliver insight into newer virtualization technologies within the cloud such as container-based virtualization, increasing adoption of serverless computing, specific accelerators, and others. We intend to develop prediction models that forecast the performance of applications on cloud servers based on workload features and instance configurations.

Cloud computing effectively manages global networking for scientific experiments and their data analysis. Furthermore, it enables a quick turnaround of resource-intensive experiments and serves as a flexible collaboration tool. In essence, it is a flexible environment that offers super-efficient remote computing resources. Cloud computing provides a flexible execution environment for scientific workloads, particularly for loosely coupled or embarrassingly parallel applications. However, for tightly coupled HPC workloads requiring low-latency communication, traditional HPC systems remain superior by approximately one order of magnitude based on the metrics reported in this study.

References

- [1] Rahul Sadhwani. Comparative cost and performance analysis of general-purpose compute instances across aws, azure, gcp, and oci. In *IEEE*, 2026. doi: 10.1109/11659657.
- [2] Pramath Parashar. Comparative analysis of distributed storage systems: Architectural design, performance, and cost trade-offs in modern cloud environments. *IJIRMP*, 13(4), 2025. doi: 10.37082/IJIRMP.v13.i4.232624.
- [3] Tahmina Anondi. Performance dynamics in big data cloud streaming systems. In *2025 IEEE Conference on Recent Advances in Artificial Intelligence (RAAI)*, 2025. doi: 10.1109/RAAI67517.2025.11423241.
- [4] Renju Rajeev. Adaptive backup for high-performance computing systems. Available at: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=AQzezQoAAAAJ&citation_for_view=AQzezQoAAAAJ:qjMakFHDy7sC.
- [5] Chintan Pamnani. Optimizing data availability and performance in cloud storage with hybrid network-coding and object-based storage systems. In *2025 IEEE International Conference on Distributed Computing and Intelligent Systems (DISCOVER)*, 2025. doi: 10.1109/DISCOVER66922.2025.11258954.
- [6] Sushant Shivankar. Modular design and performance metrics for air drives in emergency ventilator system, 2025.
- [7] Bhuvan Chandra Sarakam. High-performance real-time data store, 2025.
- [8] Tanish Gupta. High-performance deep learning for large-scale image recognition: A gpu-accelerated convolutional approach. In *IEEE*, 2025. doi: 10.1109/11409295.

- [9] V. Soundararajan, B. Agrawal, B. Herndon, P. Sethuraman, and R. Taheri. Benchmarking a virtualization platform. In *IEEE International Symposium on Workload Characterization (IISWC)*, pages 99–109, 2014.
- [10] J. P. Casazza and M. Greenfield. Redefining server performance characterization for virtualization benchmarking. *Intel Technology Journal*, 10(3), 2006.
- [11] V. Makhija, B. Herndon, P. Smith, L. Roderick, E. Zamost, and J. Anderson. Vmmark: A scalable benchmark for virtualized systems. Technical Report TR 2006-002, VMware, 2006.
- [12] R. Bonafiglia, I. Cerrato, F. Ciaccia, M. Nemirovsky, and F. Risso. Assessing the performance of virtualization technologies for nfv: A preliminary benchmarking. In *Fourth European Workshop on Software Defined Networks*, pages 67–72, 2015.
- [13] S. Wei, K. Zhang, and B. Tu. Hyperbench: A benchmark suite for virtualization capabilities. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–22, 2019.
- [14] F. Ramalho and A. Neto. Virtualization at the network edge: A performance comparison. In *IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2016.
- [15] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epc-HPC. A performance evaluation of amazon ec2 cloud computing services for scientific computing. *Cloud Computing*, pages 115–131, 2010.
- [16] Q. He, S. Garg, S. Venkugopal, and R. Buyya. Sla-based resource provisioning for cloud computing. *IEEE Transactions on Services Computing*, 2010.
- [17] Sheshukumar Vangala. Orchestrating heterogeneous storage services into a coherent virtualized data plane via middleware protocol design. *International Journal of Information Systems and Data Science (IJISDS)*, 2026.
- [18] Azizul Hakim Rafi. Unveiling customer sentiments: A comprehensive analysis of product reviews on amazon. In *EasyChair*, 2025. URL <https://easychair.org/conferences/submitted?a=33347064&submission=7013747&track=319683>.