COMP4651 Fall 2024 PROJECT

Benchmarking Performance Metrics of Cloud Computing: A Focus on CPU, Disk I/O, and Database Systems

Written by: Mohammad Taha

Student ID: 20740046

ITSC: mtaha@connect.ust.hk

Due Date: 30th November 2024

Course Instructor: Guo, Song

Department of Computer Science and Engineering

Hong Kong University of Science and Technology

Clear Water Bay, Kowloon

Hong Kong

Introduction

In the era of cloud computing, the efficiency and performance of cloud services are critical for businesses and developers alike. This report focuses on benchmarking the performance of key resources in cloud environments, specifically Amazon EC2, by evaluating CPU, disk I/O, and database operations. As organizations increasingly rely on cloud infrastructure to run applications and manage data, understanding the performance characteristics under various workloads becomes essential for optimizing resource allocation and enhancing system responsiveness. By systematically measuring and analyzing these performance metrics, this study aims to provide valuable insights into the capabilities and limitations of cloud services, ultimately guiding users in making informed decisions when configuring and utilizing cloud resources.

Method

The methodology for this project involves utilizing Amazon EC2 through the AWS Academy to conduct performance benchmarking on various instance types. Five different virtual machines will be launched, all running Ubuntu Server 22.04 LTS on a Hardware Virtual Machine (HVM) with SSD volume type. The instances selected for testing include t2.micro, t2.large, t3.medium, t3.large, and c5.large, representing a range of compute capabilities and configurations, utilizing a 64-bit x86 architecture to ensure compatibility with the benchmarking tools and workloads.

To assess the performance of these instances, a variety of benchmarking tools will be employed, including fio for disk I/O tests and pgbench for database performance evaluation. Each tool will simulate varying workloads, allowing for an analysis of how each instance performs under different conditions. By adjusting parameters such as the size of the data sets, the number of concurrent users, and the type of operations (e.g., read vs. write), a comprehensive view of the performance characteristics of each instance type can be generated.

CPU Testing

Sysbench is a versatile open-source benchmarking tool designed to evaluate the performance of system components such as CPU, memory, and I/O. Originally developed for MySQL, it supports various testing scenarios and integrates easily into environments like Amazon EC2. Its CPU benchmarking feature measures computational power through mathematical calculations, providing insights into CPU performance under different loads and enabling detailed analysis of efficiency and responsiveness.

The CPU testing will be conducted under three varying workloads to comprehensively evaluate the performance of the different EC2 instances. The first workload will involve a light computational load, where Sysbench will calculate prime numbers up to a limit of 1,000. This scenario simulates basic processing tasks that a server might encounter. The second workload will increase the complexity by raising the maximum prime number limit to 5,000, thereby testing the CPU's ability to handle more intensive calculations. The third workload will push the limits further by setting the maximum prime number to 30000, representing a high computational demand scenario. Throughout all tests, a constant number of 500 threads will be utilized to analyze the impact of concurrent processing on performance. This approach will provide a comprehensive view of how each instance type performs under different CPU load conditions.

Table 1. Benchmark Results of CPU Performance

Instance Type	Max. Prime Limit	Execution Time (s)	Throughput (ops/s)	Average Latency (ms)
t2.micro	1000	10.0100	55553.82	2.17
	10000	10.0616	2303.30	77.82
	30000	10.2710	508.60	355.70
t2.large	1000	10.0102	42899.12	2.95
	5000	10.0700	4649.20	104.27
	30000	10.0928	1040.34	466.45
t3.medium	1000	10.0148	34315.07	3.66
	5000	10.0732	3708.94	128.97
	30000	10.3256	320.40	403.57
t3.large	1000	10.0133	38032.87	12.3
	5000	10.0640	4092.05	118.58
	30000	10.8141	369.83	1304.91
c5.large	1000	10.0091	45951.69	10.45
	5000	10.0722	4911.87	98.91
	30000	10.5081	429.61	1114.44

- **t2.micro:** Highest throughput for small prime limits but performance drops significantly for larger limits.
- **t2.large**: Better overall performance compared to t2.micro, especially for larger prime limits.
- **t3.medium:** Moderate performance, with throughput decreasing as the prime limit increases.
- **t3.large**: Slightly better throughput than t3.medium but with higher latency for larger prime limits.
- **c5.large:** Outperforms all other instances in terms of throughput and latency, making it the best choice for high-performance tasks

Disk I/O Testing

Disk I/O testing is essential for evaluating the performance of a system's storage subsystem, especially in cloud environments where storage speed can significantly impact application performance. In this project, the fio (Flexible I/O Tester) tool will be used to benchmark disk I/O performance across various Amazon EC2 instances. Fio is a versatile benchmarking tool that simulates different I/O workloads, including random and sequential read and write operations. It can be configured to test various block sizes, I/O depths, and access patterns, providing comprehensive insights into disk performance characteristics.

The disk I/O testing will be conducted under three distinct workloads with a constant execution time of 60 seconds for each test. The first workload will focus on random read operations with a block size of 4 KB, simulating typical read-heavy applications. The second workload will shift to random write operations, also using a block size of 4 KB, to evaluate the instances' performance in write-heavy scenarios. Throughout all tests, a constant number of 4 jobs will be utilized to analyze the impact of concurrent I/O operations on performance.

The performance metrics for disk I/O testing will include execution time, IOPS (Input/output Operations Per Second), and throughput. Execution time measures the duration taken to complete the disk I/O benchmark, with shorter times indicating better performance. IOPS quantifies the number of read and write operations completed in one second, serving as a critical indicator of storage performance; higher IOPS values reflect better capability to handle I/O operations. Throughput measures the amount of data processed over the 60-second period, expressed in megabytes per second (MB/s), reflecting how efficiently the disk subsystem manages data transfers. By analyzing these metrics, particularly IOPS as the primary performance indicator, valuable insights can be gained regarding the efficiency and effectiveness of each EC2 instance in handling disk I/O workloads.

Table 3. Benchmark Results of Disk I/O Performances

Instance Type	Workload Type	IOPS	Throughput (MB/s)
t2.micro	Random Read	3051.21	12.5
	Random Write	18448.40	55.2
t2.large	Random Read	3110	12.7
	Random Write	31800	130
t3.medium	Random Read	3046	11.9
	Random Write	33200	136
t3.large	Random Read	3047.92	12.5
	Random Write	33100	136
c5.large	Random Read	3045	12.5
	Random Write	33200	136

- **Random Read** performance is quite similar across all instance types, with IOPS around 3000 and throughput around 12.5 MB/s.
- **Random Write** performance varies more significantly:
  - **t2.micro**: Lowest IOPS and throughput.
  - **t2.large**: Significant improvement in both IOPS and throughput compared to t2.micro.
  - **t3.medium**, **t3.large**, and **c5.large**: Similar Random Write performance, with the highest IOPS and throughput

Database Performance Testing

Database performance testing is essential for evaluating how well a system can manage database operations under various workloads. In this project, pgbench—a benchmarking tool specifically designed for PostgreSQL—will be used to assess the performance of different Amazon EC2 instances in handling mixed read and write transactions. Pgbench can simulate multiple client connections and execute a variety of predefined transactions, providing insights into the performance under realistic application scenarios.

The database performance testing will focus on mixed read and write operations, with the workload varied by using three different numbers of clients. This approach allows for an assessment of how each EC2 instance performs under varying levels of concurrent database operations, helping to understand the scalability and responsiveness of the system. The number of threads will remain constant at 2 for all tests.

The performance metrics recorded for database testing include transactions per second (TPS) and latency. Transactions per second (TPS) quantifies the number of transactions processed in one second, serving as a critical indicator of database performance; higher TPS values indicate a greater ability to handle concurrent operations. Latency measures the average time taken for individual transactions to complete, typically expressed in milliseconds, with lower latency values reflecting faster transaction processing times. By analyzing these metrics, particularly TPS as the primary performance indicator, valuable insights can be gained regarding the efficiency and effectiveness of each EC2 instance in managing database workloads.

Table 4. Benchmark Results for Database Performance

Instance Type	No. of Clients	Transactions Per Second (TPS)	Latency (ms)
t2.micro	10	1736.112920	5.760
	25	1943.058892	12.866
	50	1696.787703	29.467
t2.large	10	2229.136508	4.486
	25	2282.488022	10.953
	50	2175.656028	22.982
t3.medium	10	1941.730983	5.150
	25	1998.049625	12.512
	50	1760.618741	28.399
t3.large	10	2220.322120	4.504
	25	2250.639902	11.108
	50	1985.508331	25.182
c5.large	10	2872.291322	3.482
	25	2936.348768	8.514
	50	2690.624562	18.583

- **t2.micro**: Performance decreases as the number of clients increases, with higher latency and lower TPS at 50 clients.
- **t2.large**: Better performance compared to t2.micro, with relatively stable TPS and latency across different client counts.
- **t3.medium**: Moderate performance, with a noticeable drop in TPS and increase in latency at 50 clients.
- **t3.large**: Similar to t3.medium but with slightly better performance and lower latency.
- **c5.large**: Best overall performance, with the highest TPS and lowest latency across all client counts.

Conclusion

In this project, the performance of various AWS instance types (t2.micro, t2.large, t3.medium, t3.large, and c5.large) across CPU, disk I/O, and database operations using pgbench was evaluated. The findings are summarized as follows:

1. **CPU Performance**:

   - **c5.large** consistently outperformed other instance types, offering the highest throughput and lowest latency, making it ideal for high-performance tasks.
   - **t2.micro** showed the highest throughput for small workloads but struggled with larger ones.
   - **t2.large** and **t3.large** provided balanced performance, with **t3.large** slightly better in throughput.

2. **Disk I/O Performance**:

   - **Random Read** performance was similar across all instance types.
   - **Random Write** performance varied, with **t3.medium**, **t3.large**, and **c5.large** delivering the best results. **t2.micro** had the lowest performance.

3. **Database Performance (pgbench)**:

   - **c5.large** again led in performance, handling higher client loads with the highest TPS and lowest latency.
   - **t2.large** and **t3.large** offered stable performance, while **t2.micro** showed significant performance drops with increased client loads.

Overall, **c5.large** emerged as the most efficient instance type across all tested scenarios, making it the recommended choice for high-performance and database-intensive applications. For cost-sensitive applications with moderate performance needs, **t3.large** and **t2.large** are viable alternatives.