PROWESS

Technical Research Report

Improving Price and Performance with AWS® Spot-Based Instances

Prowess Consulting tested whether spot-based instances are a viable cost-saving strategy for running Monte Carlo workloads on Amazon Web Services® (AWS).

Executive Summary

Prowess Consulting tested on-demand and spot-based Amazon® Elastic Compute Cloud™ (Amazon EC2®) instances using the financial services workload sample with Monte Carlo European options simulation workload. Intel had earlier run Monte Carlo testing on several Amazon Web Services® (AWS®) instance types with regular on-demand subscriptions, but we wanted to go deeper with more testing. In this new study sponsored by Intel, our testing goals were to:

1. Validate the results of the Intel testing with similar tests on more instance types.
2. Expand the test scope from pure performance to price performance—that is, to measure the cost of performing the same work on different instances and processor types.
3. Explore the feasibility and price performance of using spot-based pricing—a lower pricing tier contingent on availability of excess capacity—instead of on-demand pricing.

At the time of our testing, instances based on 3rd Gen Intel® Xeon® Scalable processors emerged as top performers in terms of throughput and speed. Our testing also showed that instances based on 2nd Gen Intel Xeon Scalable processors were price performance champions.[1] While some instances based on AMD® and AWS Graviton® processors were competitive in terms of raw performance and price performance for on-demand instances, spot-based instances for these processors were often not available at the time of testing.

Comparing Amazon EC2® Spot-Based Instances

Intel®-based instances provide better raw performance and better price performance, up to:

52% faster workload completion for Intel M6i instances than for AMD EPYC™ M6a instances[2]

25% better price performance for Intel C5 instances than for AWS® Graviton3 C7g instances[3]

Intel-based C5 instances deliver up to:

2.58x better price performance using spot-based pricing than with on-demand pricing[4]

An Alternative Pricing Model to Consider

As an alternative to its standard on-demand subscription pricing model, Amazon also offers spot-based pricing for AWS, which essentially makes excess capacity available for a discounted price for as long as it remains available. AWS describes spot-based pricing as follows:

> "Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. You can use Spot Instances for various stateless, fault-tolerant, or flexible applications such as big data, containerized workloads, CI/CD, web servers, high-performance computing (HPC), and test & development workloads."[5]

The appeal of spot-based instances is the potential to significantly lower the price per hour. The actual price you pay—and your savings compared to on-demand pricing—can be complex to determine. The AWS Spot Instance advisor, for example, can tell you the current price of a particular spot-based instance, and the average savings compared to the same instance's on-demand price over the last 30 days. But the price information is unique to each region (there are 21 regions to choose from) and each specific instance type available in that region (there are 473 instance types in the US East/Ohio region, for example). With thousands of different instance type/region combinations to choose from, pricing varies.

AWS offers best practices and price-history comparison tools to help navigate the finer details of optimizing spot-instance pricing if you decide to go down that road. The purpose of this paper, however, is to address the feasibility of using spot-based instances to economically run Monte Carlo workloads on AWS.

Choosing Instances for Monte Carlo Simulation

Monte Carlo simulation is a numerical method of approximating solutions to quantitative problems by using statistical sampling techniques. Monte Carlo workloads are used in computational finance to probabilistically calculate values with multiple sources of uncertainty and randomness, such as changing interest rates, stock prices, or exchange rates, in order to evaluate complex instruments, portfolios, and investments.

Financial institutions have little guidance as to what instance types to use for running batches of Monte Carlo workloads on Amazon EC2. As a starting point, Intel ran Monte Carlo performance tests on six AWS instance types, including one based on the latest AMD processors and two based on the latest Intel® processors. The findings from Intel's testing are summarized as follows:

> "Intel completed performance testing across six instance types on Amazon Web Services (AWS) to help organizations decide which instances make the most sense to use for their Monte Carlo simulations. The results show that AWS C6i and M6i instances powered by 3rd Gen Intel Xeon Scalable processors can outperform older Intel processor–based instances, in addition to the latest M6a instances using AMD processors."[6]

It is certainly useful to know that instances based on 3rd Gen Intel Xeon Scalable processors can outperform other instance types in processing Monte Carlo simulations. But your choice of instance type is probably not going to be determined by pure performance; price performance is usually the critical factor. AWS provides data on the prices of different instance types, and Intel testing provides some data on the performance of different instance types, but we are still left with the unanswered question: which instances can complete Monte Carlo simulations at the lowest cost?

Prowess Consulting Extends Intel's Testing

Intel commissioned Prowess Consulting to extend its earlier study. Our engineers designed new testing with the following goals:

- Validate the results of the Intel testing with similar tests but on more instance types, including one based on the new AWS® Graviton3 processor.
- Expand the test scope from pure performance to also include price performance by measuring the cost of performing the same work on different instances and processor types.
- Explore the feasibility and price performance of using spot-based instance pricing instead of regular on-demand pricing.

Moreover, we sought to answer questions such as:

- How much cost savings can be realized by switching to spot-based instances?
- What is the risk of spot-based instances being unavailable to complete a workload?
- How do Intel processor–based instances compare to AMD processor–based instances and AWS Graviton processor–based instances for all these factors?

We tested the following Amazon EC2 instances using both the on-demand subscription model and the spot-based pricing model:

- C5.24xlarge (with 2nd Gen Intel Xeon Scalable processors)
- C6i.32xlarge (with 3rd Gen Intel Xeon Scalable processors)
- M6a.48xlarge (with 3rd Gen AMD EPYC™ processors)
- M6i.32xlarge (with 3rd Gen Intel Xeon Scalable processors)
- C5a.24xlarge (with 2nd Gen AMD EPYC processors)
- C7g.16xlarge (with AWS Graviton3 processors)

There are different options for what happens when there's an interruption to spot-based instances. We chose hibernation, so that any delays will be quantifiable as performance degradations. There might be different pricing/availability for spot-based instances at different times and in different geographies, so we tested at different times in different AWS locations to see if any patterns emerged.

We collected the following key data points during the testing process:

- Total duration of the on-demand test run
- Total duration of the spot-based test run
- Actual cost of the on-demand and spot-based test runs
- AWS region of the on-demand and spot-based instance testing
- Incidents of unavailability of spot-based instances

The remainder of this paper outlines the results of our testing.

Test Results

Prowess Consulting evaluated on-demand and spot-based instances to compare both raw performance and price performance. Results, where normalized, use the C5.24xlarge instance as a baseline. That widely used instance type is based on 2nd Gen Intel Xeon Scalable processors, so using it as a baseline provides easy visualization of performance differences with newer instance types based on 3rd Gen Intel Xeon Scalable processors, in addition to those based on non-Intel processors.

Raw Performance

We measured raw performance as the time it took the different instances to complete the Monte Carlo tests. The top performers were the M6i and C6i instances based on 3rd Gen Intel Xeon Scalable processors. This validates earlier test results by Intel comparing on-demand instances based on different processors.[8]

In each case, there was no discernable difference in raw performance between the on-demand and spot-based instances. This was somewhat contrary to our original expectations; we initially anticipated the spot-based instances would sometimes hibernate when resources became unavailable, and they would therefore take longer to complete the tests.

However, as it turns out, the instances we tested do not support such hibernate-and-resume scenarios. Therefore, when test runs were interrupted for lack of resources, they simply stopped running and the tests did not run through completion. These events did not affect the performance results, which were based on tests that completed successfully. (For a discussion of our experience with instance unavailability, see the Availability Issues section.)

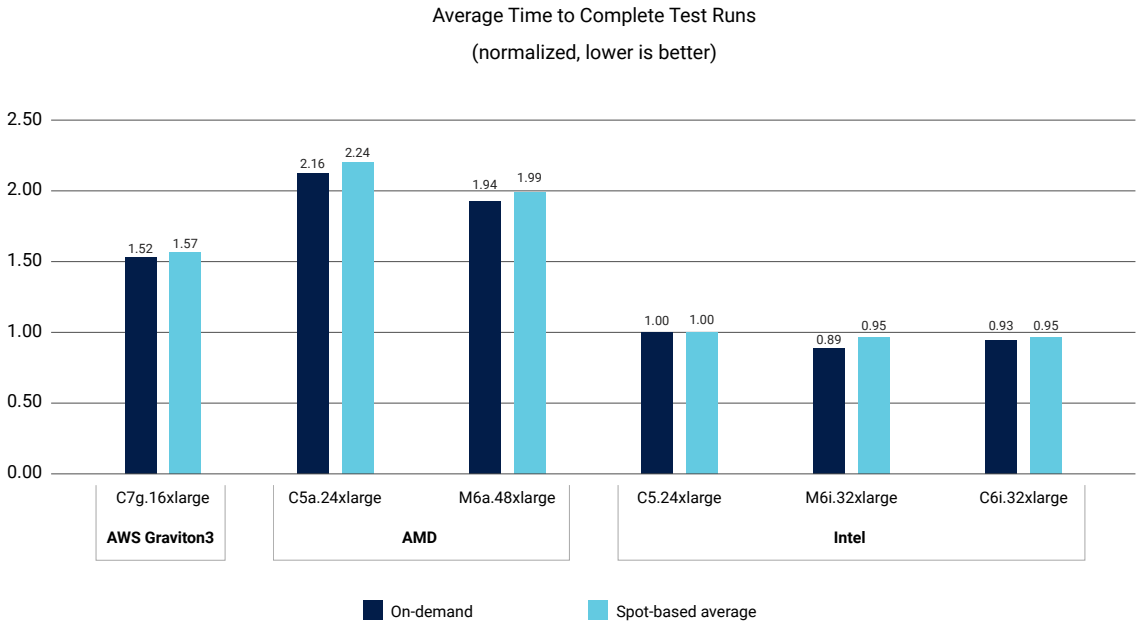

Figure 1 | Performance measured the time needed to complete Monte Carlo test runs[1]

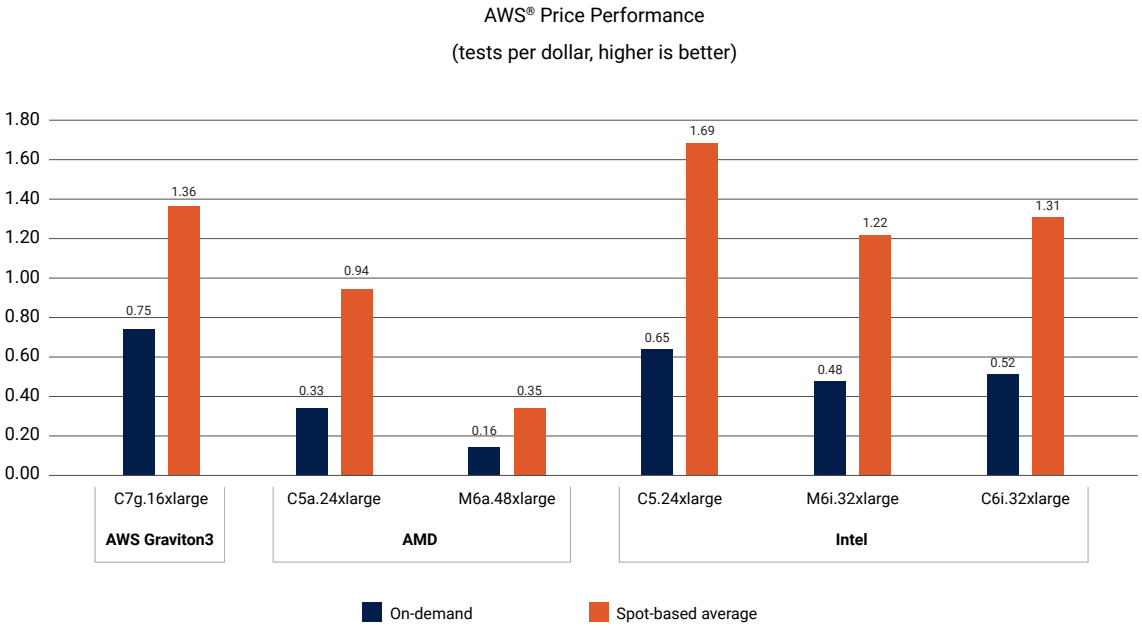

Figure 2 | Spot-based instances performed more tests per dollar, on average, than the same instances on-demand[1]

Figure 3 shows the same data with price performance for spot-based instances broken out by location, as the price of instances can vary by location.

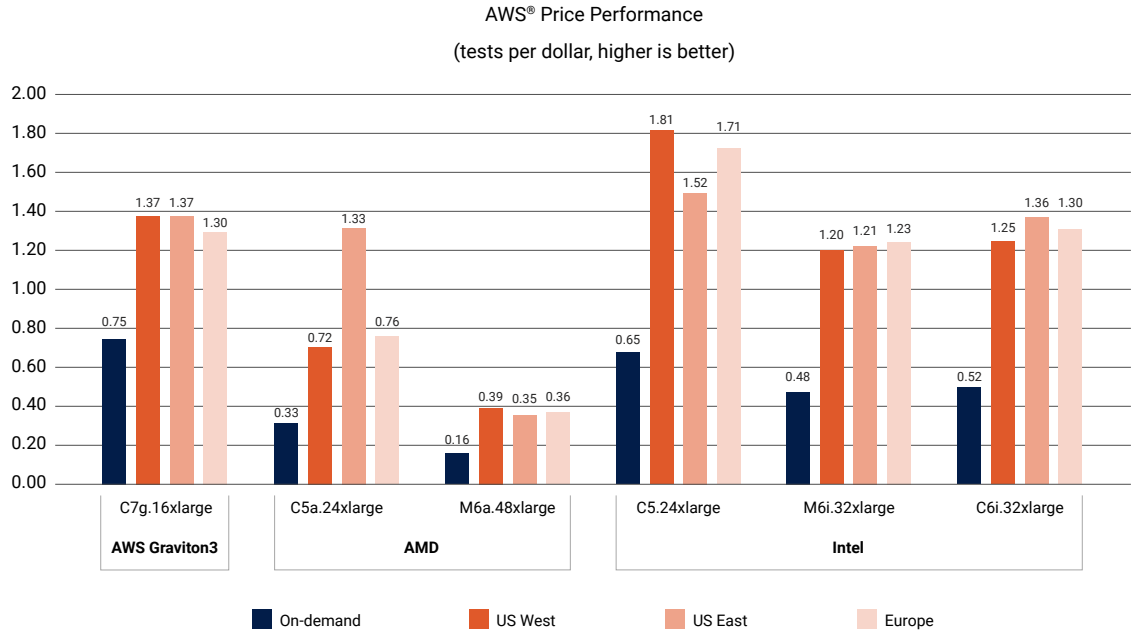

Figure 3 | Comparing price performance between on-demand instances and spot-based instances in three regions[1]

Availability Issues

By their nature, spot-based instances are not guaranteed to be available when and where you want them—that's why they cost less. A key consideration for anyone considering adopting spot-based instances as a cost-saving strategy is how available the instances will be. Our testing does not provide a conclusive answer to that question, but it does provide some insight into what kind of availability issues one might expect when moving to spot-based instances.

During the course of testing, we attempted to run Monte Carlo tests on each of six instance types, in three different locations, and on three different days. In other words, we attempted 54 test runs on spot-based instances. In 17 attempts out of 54 (31 percent), the test run could not be completed due to lack of availability. This 69 percent success rate, however, was not evenly distributed among instance types, as shown in Figure 4.

AWS® Spot-Based Instance Availability

		C5.24xlarge **Intel**	C6i.32xlarge **Intel**	M6i.32xlarge **Intel**	M6a.48xlarge **AMD**	C5a.24xlarge **AMD**	C7g.16xlarge **AWS Graviton3**
Day 1	US East (N. Virginia)	+	+	+	−	−	+
	US West (Oregon)	+	+	+	+	+	+
	Europe (London)	+	+	+	−	+	−
Day 2	US East (N. Virginia)	+	+	+	−	−	+
	US West (Oregon)	+	+	+	−	−	+
	Europe (London)	+	+	+	−	+	−
Day 3	US East (N. Virginia)	+	+	+	−	−	−
	US West (Oregon)	+	+	+	−	−	+
	Europe (London)	+	+	+	−	+	−
	% Available	**100%**	**100%**	**100%**	**11%**	**44%**	**56%**

Figure 4 | Availability of six spot-based instances on three days and in three locations

The most striking pattern in this availability sampling is that our test engineers encountered no availability issues at all in completing the 27 tests on Intel processor–based instances, but they were blocked by unavailability in 17 of the 27 instances based on other processors—only a 37 percent success rate.

These results, while not conclusive, support the hypothesis that the spot-based pricing model is more available for Intel-based instances than for instances using AMD EPYC or AWS Graviton processors.

Analysis and Conclusions

Tests conducted by Prowess Consulting support the following conclusions:[1]

- In terms of raw performance, measured as time to completion, the clear winners are instances based on 3rd Gen Intel Xeon Scalable processors.
- There is virtually no performance difference between demand-based and spot-based instances. The only issue is the availability of spot-based instances.
- In our experience, spot-based instances with Intel processors were readily available, while spot-based instances with AMD EPYC or AWS Graviton processors were available less than half the time.
- Price performance differed widely between demand-based and spot-based instances, with spot-based instances in every case showing as the more cost-effective option.
- The best price performance overall was achieved by the C5.24xlarge instance based on 2nd Gen Intel Xeon Scalable processors. While this instance did not complete the tests as quickly as the instances based on 3rd Gen Intel Xeon Scalable processors, the lower prices made it a stand-out option.
- Comparing on-demand instances, the C5.24xlarge instance again delivered the best price performance of any widely available instance type. (Though the C7g.16xlarge showed somewhat better results, this next-gen instance based on AWS Graviton3 processors is not yet widely available.)

Spot-based instances on AWS represent a fertile area to hunt for cost-saving opportunities when running Monte Carlo workloads. Spot-based instances deliver the same performance as on-demand instances, but at a substantially lower price. When choosing a spot-based instance type, price performance is an important consideration, and Intel instances deliver the best price performance of the options we tested. Other factors to consider include the availability of spot-based instances of the type you choose, the differences in pricing at different AWS locations, and the ability of certain instances to hibernate and resume a workload while other instances can only stop and fail if availability disappears. Using spot-based instances for Monte Carlo workloads is feasible and can reduce costs, but there is a learning curve to using this option in the most efficient manner.

Learn More

Learn more about Amazon EC2 spot-based pricing at https://aws.amazon.com/ec2/spot/.

Read the Intel solution brief, “Boost Monte Carlo Simulations with Newer Amazon Web Services C6i and M6i Instances Featuring 3rd Gen Intel® Xeon Scalable Processors.”

Appendix: Test Details

Prowess Consulting engineers conducted the testing by performing the following steps:

1. Create the on-demand or spot-based instance.
    a. Conduct the spot-based instance testing at different times of day and at different AWS locations.
    b. Set the spot-based instance interrupt to hibernate.
    c. Create an Amazon® CloudWatch topic and an Amazon® Simple Notification Service (SNS) notification to send an alert when the spot-based instances are interrupted.
2. Use CentOS® 8 for the operating system (OS).
3. Install the following compiler and libraries:
    a. Intel® oneAPI HPC Toolkit (HPC Kit): https://registrationcenter-download.intel.com/akdlm/irc_nas/18679/l_HPCKit_p_2022.2.0.191_offline.sh
    b. Intel® oneAPI Base Toolkit (Base Kit): https://registrationcenter-download.intel.com/akdlm/irc_nas/18673/l_BaseKit_p_2022.2.0.262_offline.sh
4. Set the required parameters:
    a. export PR=<*number of active cores*>
    b. export OMP_NUM_THREAD=1
5. Build the Monte Carlo workload.
6. Run the Monte Carlo workload:
    a. Intel processor–based Amazon EC2 instances:
        - ./runbatch.sh MonteCarloInsideBlockingDP.avx512
    b. AMD processor–based Amazon EC2 Instances:
        - ./runbatch.sh MonteCarloInsideBlockingDP.avx2

[1] Based on testing by Prowess Consulting as of October 2022. For configuration details, see Prowess Consulting Extends Intel's Testing; for test results, see Test Results; and for test details, see Appendix: Test Details.

[2] Based on throughput measured as number of Monte Carlo options per second, comparing 2nd Gen Intel® Xeon® Scalable processor–based M6i.32xlarge instances vs. 3rd Gen AMD EPYC™ processor–based M6a.48xlarge spot-based instances.

[3] Price performance measured as number of Monte Carlo tests completed per dollar, comparing 2nd Gen Intel® Xeon® Scalable processor–based C5.24xlarge instances vs. AWS® Graviton3 processor–based C7g.16xlarge spot-based instances.

[4] Price performance measured as number of Monte Carlo tests completed per dollar for 2nd Gen Intel® Xeon® Scalable processor–based C5.24xlarge instances, comparing spot-based pricing vs. on-demand pricing.

[5] AWS. "Amazon EC2 Spot Instances." Accessed September 2022. https://aws.amazon.com/ec2/spot/.

[6] Intel. "Boost Monte Carlo Simulations with Newer Amazon Web Services C6i and M6i Instances Featuring 3rd Gen Intel® Xeon® Scalable Processors." [URL TBD]

0723/220125