Intel® Xeon® Processor-Based Instances Achieved 1.2x as Much Java Web Services Work as Instances with ARM-Based Processors

Java

Handle 1.2x as much Java Web Services work with 64-vCPU instances with 2nd Gen Intel Xeon Scalable processors

vs. ARM-based instances

Avoid the extra investment that can be necessary when changing from one CPU architecture to another

Get Greater Performance by Selecting Instances Featuring 2nd Gen Intel Xeon Scalable Processors

If your company runs Java web services workloads on AWS, you want to be sure to select the right instance type for the job. One consideration is performance, and another is continuity. By choosing instances enabled by 2nd Gen Intel Xeon Scalable processors rather than instances powered by ARM-based processors, you could reap advantages in both areas.

Testing with an Industry-standard Java server benchmark compared two extra-large instances with 64 vCPUs: instances enabled by 2nd Gen Intel Xeon Scalable processors and instances powered by ARM-based processors. The instances featuring Intel Xeon processors did 1.2x the work of the ARM processor-based instances. These performance gains could translate to needing fewer instances to do a given amount of work.

Assuming you already run your Java web services workloads on VMs powered by 2nd Gen Intel Xeon Scalable processors, selecting instances featuring the same processors can not only boost performance, but can provide a more seamless and predictable experience. Switching from Intel Xeon architecture to ARM-based architecture could require you to re-optimize your workload, among other things.

For your Java web services needs, choose instances enabled by 2nd Gen Intel Xeon Scalable processors.

Better Performance per Instance Can Reduce the Number of Instances You Need

If you've decided to host your Java web services workloads on the cloud, you can get better performance by selecting instances enabled by 2nd Gen Intel® Xeon® Scalable processors rather than instances based on ARM-based processors. As Figure 1 shows, an extra-large 64-vCPU instance enabled by Intel Xeon Scalable processors achieved 1.2x the Java work.

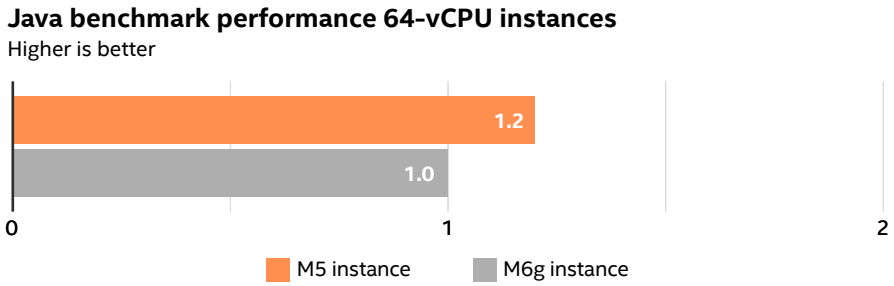

Figure 1. Relative test results comparing the Java web services performance of the extra-large (64-vCPU) M5 instance type to the extra-large M6g instance type.

Sticking With a Known Processor Architecture Can Improve Continuity and Reduce Extra Work

If you've been running your Java web services workloads on Intel processors, you already have a set of baseline settings and best practices in place. You have likely optimized your workloads for that processor architecture. Changing to a new ARM-based processor architecture can require additional investments of time: your old assumptions and best practices might not apply. You'd have to perform testing, and might also have to tweak your workloads to get the best possible performance. By choosing instances enabled by 2nd Gen Intel Xeon Scalable processors, you avoid all this extra work.

Learn More

To begin running your Java services workloads on instances with 2nd Gen Intel Xeon Scalable processors, visit http://intel.com/aws.

Tests performed by Intel in October 2020 on AWS. Tested three iterations and selected the median for result. Software used was Ubuntu 20.04 with kernel 5.4.0. Other software included Java Server Benchmark with openjdk version 11.0.2 for Intel instances and AdoptOpenJDK version 11.0.7 for Graviton instances. Both configurations used EBS storage; other configuration details to follow. m5.16xlarge: 64 vCPUs, 256GB RAM, Intel 8259CL CPU, and 2 workload groups. m6g.16xlarge: 64 vCPUs, 256GB RAM, Amazon Graviton 2 CPU, and 1 workload group.

Printed in USA 0921/JO/PT/PDF US002 ♻ Please Recycle