Article

Murray Oldfield · Nov 10, 2022 7m read

Using an LVM stripe to increase AWS EBS IOPS and Throughput

Overview

Predictable storage IO performance with low latency is vital to provide scalability and reliability for your applications. This set of benchmarks is to inform users of IRIS considering deploying applications in AWS about EBS gp3 volume performance.

Summary

- An LVM stripe can increase IOPS and throughput beyond single EBS volume performance limits.
- An LVM stripe lowers read latency.

> **Read First!** A reference architecture, including a more detailed explanation of configuring LVM in AWS, is here:
> https://community.intersystems.com/post/intersystems-iris-example-refere...

Need to know

gp3 EBS volume performance summary

Cloud vendors enforce storage performance limits, for example, IOPS or throughput, usually by increasing latency, which will impact application performance and end-user experience. With gp3 volumes, base-level storage performance limits can be increased by paying higher prices to increase predefined limits.

> Note that you must increase IOPS **and** throughput to get maximum performance from an EBS volume. Latency is applied when either limit is hit.

IOPS performance

- gp3 volumes deliver a consistent baseline IOPS performance of 3,000 IOPS, which is included with the storage price.
- You can provision additional IOPS up to a maximum of 16,000 IOPS per volume.
- Maximum IOPS can be provisioned for volumes 32 GiB or larger.
- gp3 volumes do not use burst performance.

Throughput performance

- gp3 volumes deliver a consistent baseline throughput performance of 125 MiB/s, which is included with the storage price.
- You can provision additional throughput (up to a maximum of 1,000 MiB/s) for an additional cost at a ratio of 0.25 MiB/s per provisioned IOPS.
- Maximum throughput can be provisioned at 4,000 IOPS or higher and 8 GiB or larger (4,000 IOPS x 0.25

MiB/s per IOPS = 1,000 MiB/s).

For more information, see General Purpose SSD volumes - Amazon Elastic Compute Cloud.

Instance limits

- EC2 instance types have maximum IOP and throughput limits. Latency is applied when instance or EBS limits are hit.

For more information, see: Amazon EBS–optimized instances - Amazon Elastic Compute Cloud

Logical Volume Manager (Linux)

If your application requires more than 16,000 IOPS from a single filesystem, you can configure multiple volumes in a Logical Volume Manager (LVM) stripe.

For example: if you require 80,000 IOPS, you can provision an EC2 instance type that can support 80K IOPS using five LVM striped gp3 volumes.

> LVM was used for all the following tests using single or multiple EBS volumes.

Other block storage types

While there are other volume types, such as io2 block express with higher IOPS and throughput for a single volume, it can be far cheaper to use gp3 EBS volumes in an LVM stripe for the same amount of storage and "high enough" IOPS. io2 block express is only of value on certain instance types.

Tests Run

Two tests are run to generate simulated IRIS application storage IO. RANREAD and RANWRITE.

Details of the tests and how to run them are here:
https://community.intersystems.com/post/perftools-io-test-suite

RANREAD

Generates random reads of a single IRIS database.

- In IRIS, reads are continuous by user processes; a user process initiates a disk IO to read the data. Daemons serving web pages, SQL queries, or direct user processes perform reads.
- RANREAD processes read random data blocks in the test database.
- The test database is sized to exceed the expected read cache of on-premises storage systems. It is unclear if AWS uses read caching (for example, Azure does it by default).

RANWRITE

Generates write IO using the IRIS database Write Daemon Cycle. IRIS's three main write activities are:

- **Write Image Journal (WIJ).** WIJ writes a burst approximately every 80 seconds or when a percentage of the database cache is pending updates by a single database master write daemon. The WIJ protects physical database file integrity from system failure during a database write cycle. Writes are approximately 256KB each immediately before the database files are updated with random database writes.

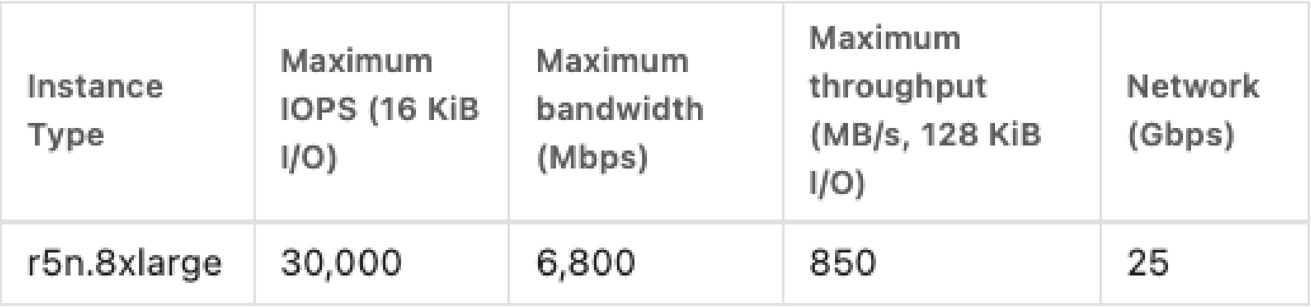
- **Random database writes.** Write daemon database writes are a burst approximately every 80 seconds or based percentage of database cache pending updates. A set of database system processes known as write daemons perform writes. User processes update the database in-memory cache, and a trigger (based on a time or activity threshold) sends the updates to disk using the write daemons. Typically anywhere from a few MBs to several GBs is written during the write cycle, depending on transaction rates.
- **Journal writes.** Journal writes are near-continuous, Less than every two seconds, triggered on full journal buffers or a data synchronisation request (for example, from a transaction). Journal writes are sequential and variable in size from 4KB to 4MB. There can be as low as a few dozen writes per second to several thousand per second for large deployments using distributed application servers.

Note: separate databases are used for RANREAD and RANWRITE in the tests; however, the same EBS volume and filesystem (/data) are used.
The RANREAD database size is 700 GiB
IRIS on Linux systems uses direct IO for database operations. Linux file system page cache is not used.

Benchmark results

A. No LVM stripe

The benchmark was run using the following EBS-optimised instance:



Instance Type	Maximum IOPS (16 KiB I/O)	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Network (Gbps)
r5n.8xlarge	30,000	6,800	850	25

The IRIS benchmark is running with **three EBS volumes (Data, WIJ, and Journals).** IOPS for the /data volume have been provisioned at the maximum for a single volume. Throughput has been left at the default of 125 MB/s.

A.002 - 30 minute RANREAD only with a stepped increase in IOPS.

Five and then ten RANREAD processes were used to generate the IO. Approaching maximum throughput (16K IOPS and 125 MB/s throughput).

> Note: IRIS is using an 8KB database block size (1,500 * 8KiB = approximately 123 MiB/s)

In the image below: The baseline latency (green) is consistent between RANREAD runs, averaging around 0.7-1.5ms, although there are peaks over 5 ms. For an unknown reason, performance degrades during the middle of the 10-process test.

A.006 - 20 minutes RANREAD and RANWRITE.

The image below shows how the test was paced for peaks around 8,000 write and 12,000 read IOPS. The gap in reads is intentional.

The read latency in the image below is similar to the A.002 test. Read and write databases are on the same EBS volume. There is a spike in the read latency (green) during the write daemon cycle random database writes (blue). The spike in the read latency is transient and probably will not affect an interactive user's experience.

> The WIJ is on a separate EBS volume to isolate the sequential writes of the WIJ from the read IOPS. Although in the cloud all EBS storage is network storage, there is no guarantee that individual EBS volumes are in separate physical SSD storage or that different volumes attached to an instance are in the same storage enclosure.

B. Increase IOPS and throughput using an LVM stripe

The benchmark suite run using the an EBS-optimised instance with a maximum of 80,000 IOPS:

The IRIS benchmark is running with five EBS volumes. Note /data comprises five EBS volumes in an LVM stripe.

B.080 - 30 minute RANREAD only stepped increase in IOPS.

The test was run with 10 to 50 RANREAD processes, stepping up in increments of 10 processes. The test is approaching the maximum IOPS (80K).

DB Read Latency is lower (average 0.7) than the no-stripe test.

B.083 - 30 minutes RANREAD and RANWRITE.

This test was paced at peaks around 40,000 write and 60,000 read IOPS.

> Note that the combined read and write IOPS is higher than the maximum IOPS of the EC2 instance (80,000 IOPS).

Read latency is similar to the B.080 test. Read and write databases are on the same EBS volume. There is a spike in the read latency (green) during the write daemon cycle random database writes (blue). The spike in the read latency is transient and less than 1.4ms and probably will not affect user experience.

#AWS #Cloud #Deployment #System Administration #Tips & Tricks #Other

Source URL:https://community.intersystems.com/post/using-lvm-stripe-increase-aws-ebs-iops-and-throughput