Aerospike Scales with Google Cloud Platform

PERFORMANCE TEST SHOW AEROSPIKE SCALES ON GOOGLE CLOUD

Aerospike is an In-Memory NoSQL database and a fast Key Value Store commonly used for caching and by real-time big data driven applications that personalize the consumer experience with consistently fast reads and writes. Google is known for scale and Google Compute Engine (GCE) is operated by Google engineers within Google’s high performance infrastructure. Developers can quickly spin up a 2 node Aerospike cluster using Google’s Click-to-Deploy technology. It was natural for us to want to test the performance of Aerospike on different types of GCE instances.

This paper describes several performance tests with Aerospike on Google Compute Engine that show dramatic price/performance that will fuel an entirely new category of applications that must process data in real-time from the very start, enabling a new class of startups with business models that were previously not economically viable.

An in-memory database with a hybrid approach, Aerospike can be configured to run in one of two storage modes:

- Data stored completely in RAM, with optional persistence on disk
- Hybrid mode with indexes in RAM and data on SSDs, which gives much better price/performance
Developers commonly use Aerospike as a server side cookie store, user profile store and id-mapping store in real-time bidding platforms, for dynamic pricing, recommendations, fraud detection and real-time analytics. With Aerospike, applications can scale far more simply and cost effectively with fewer servers and SSDs instead of RAM. In addition, Aerospike is easier to operate with self-managing features like auto-clustering, auto-sharding and auto-rebalancing.

We document results that demonstrate:

1. **One Million Writes Per Second with 6x Fewer Nodes on Google Compute Engine with Aerospike compared to Cassandra**
   - High Throughput for both Reads and Writes
     - 1 Million Writes per Second with just 50 Aerospike servers
     - 1 Million Reads per Second with just 10 Aerospike servers
   - Consistent low latency, without jitter for both reads and writes
     - 7ms median latency for Writes with 83% of writes < 16ms and 96% < 32ms
     - 1ms median latency for Reads with 80% of reads < 4ms and 96.5% < 16ms
     - Note that latencies are measured on the server, latencies on the client will be higher
   - Unmatched Price / Performance for both Reads and Writes
     - 1 Million Writes Per Second for just $41.20/hour
     - 1 Million Reads Per Second for just $11.44/hour

2. **RAM-like performance and 15x Storage Cost Advantage with Local SSDs on Google Compute Engine**
   - Storage using RAM is the standard for performance. Aerospike database using GCE SSD instances demonstrates very comparable latency at a fraction of the price
   - Write speeds at the same level as RAM

**ONE MILLION WRITES PER SECOND WITH 6X FEWER NODES ON GOOGLE COMPUTE ENGINE WITH AEROSPIKE COMPARED TO CASSANDRA**

Apache Cassandra is an open source, columnar NoSQL database that is great for ingesting and analyzing hundreds of terabytes of data stored on rotational disks. Like Cassandra, Aerospike is also an open source NoSQL database, but unlike Cassandra, Aerospike is an in-memory database that can run in pure RAM and in a hybrid mode with RAM and SSDs.

In addition, Cassandra is most often deployed with delayed consistency, while data in Aerospike is by
default, replicated synchronously in-memory to ensure immediate consistency and asynchronously written to disk.

After Google published a blog post “Cassandra Hits One Million Writes Per Second on Google Compute Engine”: http://googlecloudplatform.blogspot.com/2014/03/cassandra-hits-one-million-writes-per-second-on-google-compute-engine.html - using 300 nodes, we followed the same steps and documented how Aerospike Hits One Million Writes Per Second With Just 50 Nodes On Google Compute Engine - a 6x cost savings - while at the same time, delivering consistent low latency with no jitter for both read and write workloads.

Our benchmark used the same setup: 100 Million records at 200 bytes each, debian 7 backports, servers on n1-standard-8 instances with 500GB non-SSD persistent disks at $0.504/hr, clients on n1-high-cpu-8 instances at $0.32/hr, and followed these steps: https://github.com/aerospike/aerospike-gce-benchmark/blob/master/gce_cli_steps.txt

In addition to pure write performance, we also documented pure read and mixed read/write performance.
High Throughput, 1 Million Reads Per Second with 10 servers

This graph shows Aerospike throughput at different read/write ratios using 10 server nodes and 20 client nodes. For all read/write ratios, 80% of TPS in this graph is achieved with 50% of the clients (10), adding more clients only marginally improves throughput.

GCE disk IOPS depend on size (see https://cloud.google.com/compute/docs/disks#pdperformance). We used 500GB non-SSD persistent disks to ensure high IOPS, so the disk is not the bottleneck. For larger clusters, 500GB is a huge over-allocation and can be reduced for lower costs. To achieve this high performance, we did not need to use SSD persistent disks to get higher IOPS.

Low Latency: For a 100% read load, only ~20% of reads were >4ms and only ~3.5% took >16ms

This graph shows consistent low latency for different read/write ratios using 10 server nodes and 20 client nodes.

For a 100% read load, only ~20% of reads took more than 4ms and ~3.5% reads took more than 16ms. This is because reads in Aerospike are only 1 hop (network round trip) away from the client, while writes take 2 network-roundtrips for synchronous replication. Even with a 100% write load, only 16% of writes took more than 32ms.

We ran Aerospike on 7 of 8 cores on each server node. Leaving 1 core idle helped latency; if all cores were busy, network latencies increased. Latencies are as measured on the server.

Linear Scalability for both Read and Write workloads

This graph shows linear scalability for 100% reads and writes but you can expect linear scaling for mixed workload too.
For reads, a 2:1 clients:server ratio was used i.e for a 6 node cluster, we used 12 client machines to saturate the cluster.

For writes, a 1:1 client:server ratio was enough because of the lower throughput of writes.

**ONE MILLION READS PER SECOND ON GOOGLE COMPUTE ENGINE WITH 25X MORE CAPACITY AND 15X LOWER COSTS WITH LOCAL SSDS INSTEAD OF RAM**

We repeated the benchmark using Local SSDs instead of RAM to perform 1 Million Reads Per Second. The following servers and storage configurations demonstrate 25x more capacity and 15x lower costs for local SSDs compared to RAM.

<table>
<thead>
<tr>
<th></th>
<th>10 Servers (With RAM)</th>
<th>10 Servers (with SSD)</th>
<th>20 Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machine</strong></td>
<td>n1-standard-8</td>
<td>n1-standard-8-2x-ssd</td>
<td>n1-highcpu-8</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>30 GB of RAM</td>
<td>30 GB of RAM + 375 GB * 2 = 750 GB of SSD</td>
<td>7 GB of RAM</td>
</tr>
<tr>
<td></td>
<td>(25 x more capacity than RAM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cost per node per Month</strong></td>
<td>$254</td>
<td>$417.5 ( $254 + $81.75 * 2 ssds)</td>
<td>$162</td>
</tr>
<tr>
<td></td>
<td>(1.64 x cost of RAM)</td>
<td>(15x lower cost than RAM)</td>
<td></td>
</tr>
<tr>
<td><strong>Cost per GB per Month</strong></td>
<td>$8.46</td>
<td>$417.50/ 750GB = $0.56</td>
<td></td>
</tr>
</tbody>
</table>

**Local SSDs on GCE Show The Same Write Throughput of RAM**

The chart shows that the write performance of Aerospike on SSD is very comparable to that of RAM. At 100% write, the SSDs are able to sustain 226,000 transactions per second compared to 239,000 for RAM. In addition, the read rates sustain very high performance as well. However, as data volumes grow, the cost advantages to using SSDs grow as well.
GCE Local SSDs Delivered Approximately the Same Latency of RAM

Many may wonder how the latencies of GCE local SSDs compare to that of RAM. The chart below shows that in 2 different key thresholds (1 ms and 8ms), the percentage of transactions that exceed those time limits are roughly comparable. Although RAM is likely to have better latencies in sub-millisecond time intervals, few applications need that level of latency.

CONCLUSIONS

A new generation of applications with mixed read/write data access patterns sense and respond to what users do on websites and on mobile apps across the Internet. These applications perform data writes with every click and swipe, make decisions, record, and respond in real-time. They require a database proven to handle the scalability, throughput, and latency requirements.

On Google Compute Engine, Aerospike scales in pure RAM with 6x fewer servers than Cassandra and Aerospike with local SSDs on Google Compute Engine demonstrates 25x more capacity and 15x cost savings over Aerospike in pure RAM - a new level of price/performance that will fuel a new category of applications and accelerate a new class of businesses that will easily scale on the Internet.

TRY IT NOW!