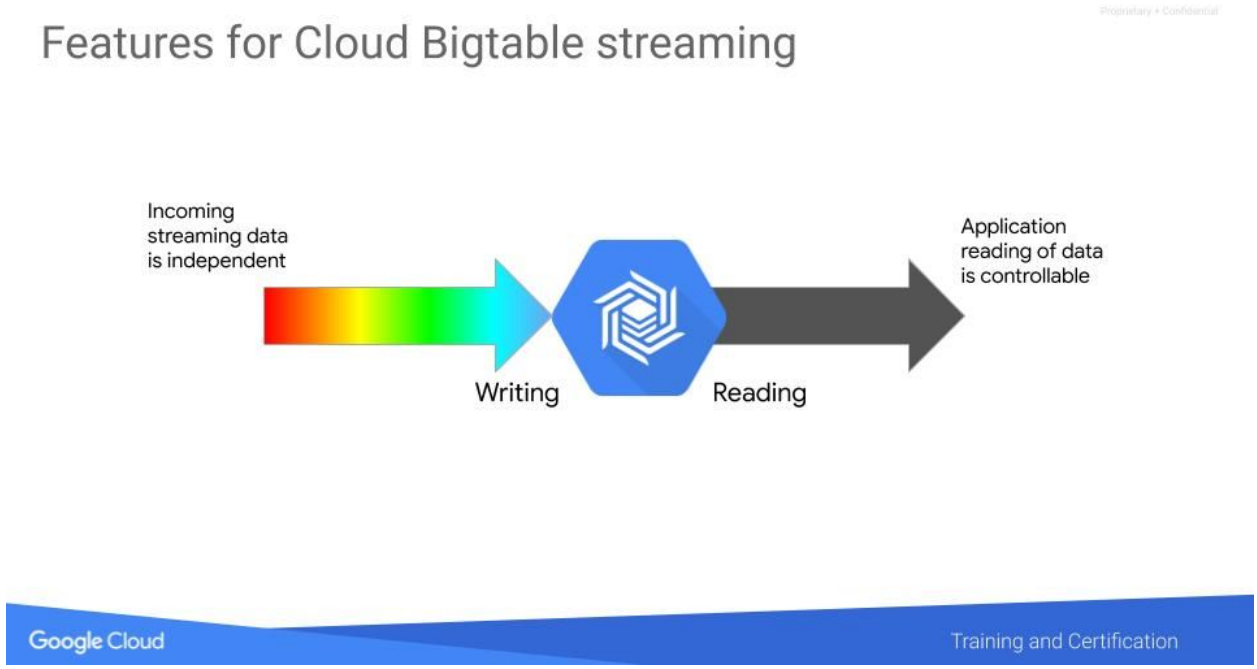
Cloud Bigtable Streaming



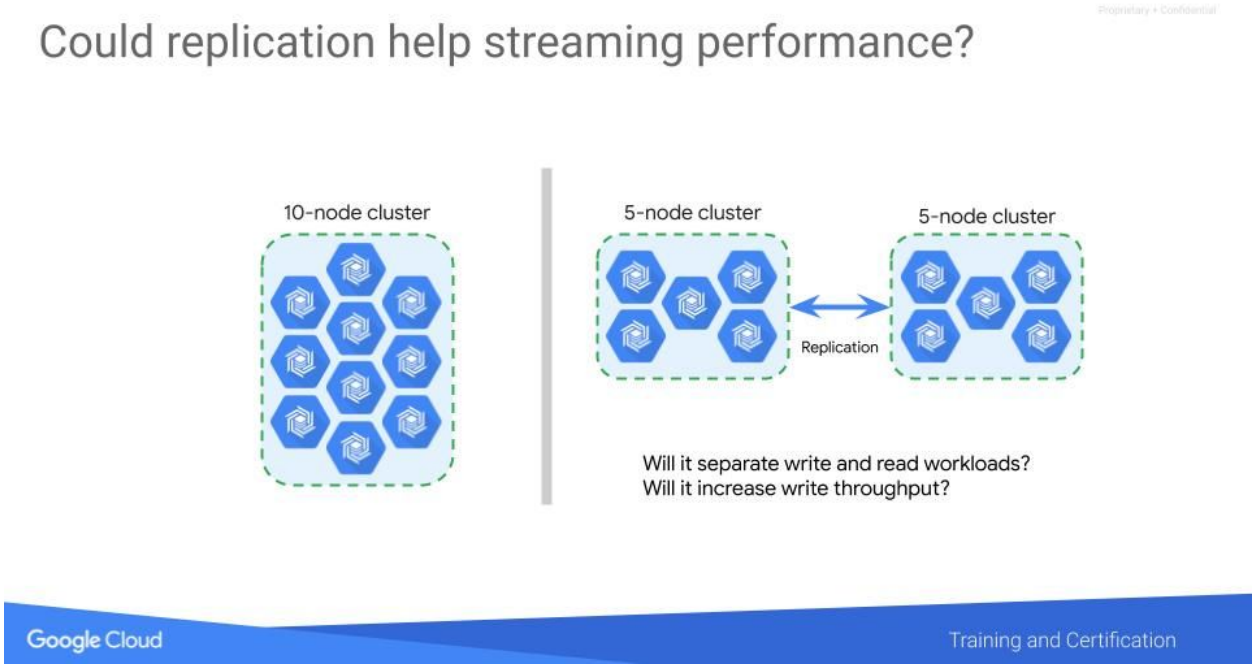
There are some things you can do that will improve write performance and read performance.
There are some things you can do that are a trade-off where improving write may cost you on read

Separate writing from reading -- write-only workloads

There are some things you can do that will improve write performance and read performance.
There are some things you can do that are a trade-off where improving write performance may cost you on read performance.

Separate writing from reading in your application to approximate write-only workloads.

https://pixabay.com/illustrations/hand-pencil-holding-sketch-drawing-1515895/



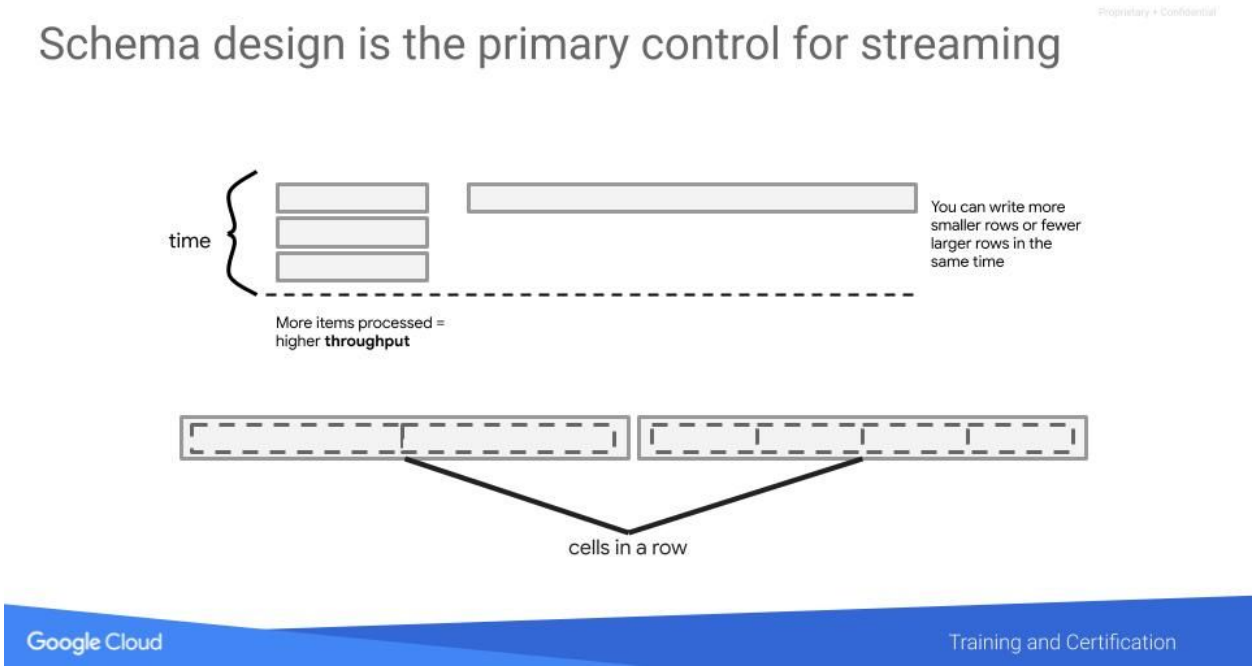
If you have a 5-node cluster and you want to add 5 more nodes, are you better off creating a 10-node cluster in a single zone or setting up replication between two 5-node clusters?

The initial thinking is that separating writing from reading through replication might help performance. In other words, the first cluster would be used only for capturing streaming data and the replica cluster would be used only for reading that data. This does not work because the replica cluster introduces read operations onto the source cluster. So it is no longer just writing the data but also it is being read. And it is a guaranteed worst case because all of the data is copied to the replica. At the same time, the read performance on the replica is now mixed with the write operations necessary to keep it updated. So the attempt to separate the read and write workloads actually does the opposite.

The next thought might be that replication will distributed the work and create better throughput. In other words, both clusters perform writes. And perhaps the additional capacity of the combined clusters will outweigh the overhead of replicating between them. This is also generally not true. Because the replication itself is another workload taxing both cluster resources.

If the replica cluster is in a different zone or region from the source, replication improves availability. But it also introduces latency in the copying process.

So... the final conclusion is that in general, you are better off for streaming performance to add additional nodes to a single cluster in a single zone, rather than introducing replication.
Remember that if you add nodes, it could take Cloud Bigtable about 20 minutes to rebalance the load before it shows a change in performance.



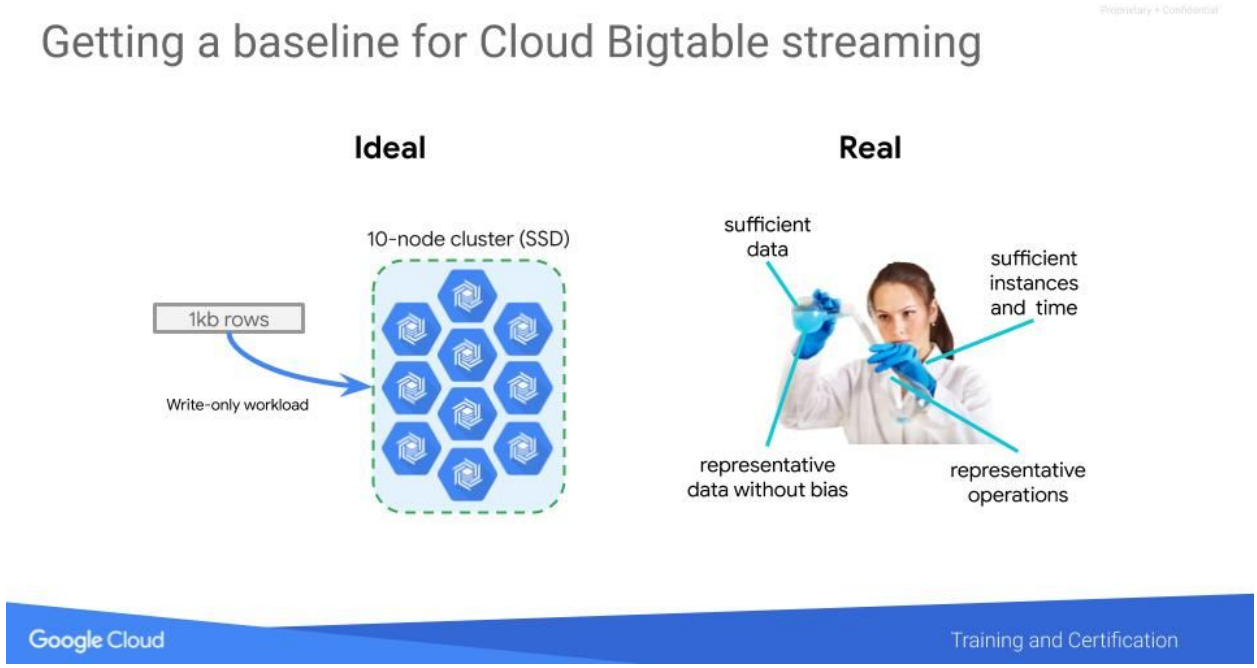
A higher throughput means more items are processed in a given amount of time.
If you have larger rows, then fewer of them will be processed in the same amount of time.
In general, smaller rows offers higher throughput, and therefore is better for streaming performance.

Cloud Bigtable takes time to process cells within a row.
So if there are fewer cells within a row, it will generally provide better performance than more cells.

Finally, selecting the right row key is critical. Rows are sorted lexicographically.
The goal when optimizing for streaming is to avoid creating hotspots when writing, which would cause Cloud Bigtable to have to split tablets and adjust loads.
To accomplish that, you want the data to be as evenly distributed as possible.

https://cloud.google.com/bigtable/docs/performance

reading delay + processing delay = response time



The generalizations, of isolate the write workload, increase number of nodes, and decrease row size and cell size will not apply in all cases.
In most circumstances, experimentation is the key to defining the best solution.

A performance estimate is given in the documentation online for write-only workloads.
Of course, the purpose of writing data is to eventually read it, so the baseline is an ideal case.

At the time of this writing, a 10-node SSD cluster with 1kb rows and a write-only workload can process 10,000 rows per second at a 6ms delay.
This estimate will be affected by average row size, the balance and timing of reads distracting from writes, and other factors.

You will want to run performance tests with your actual data and application code. You need to run the tests on at least 300 GB of data to get valid results.
Also, to get valid results your test needs to perform enough actions over a long enough period of time to give Cloud Bigtable the time and conditions necessary to learn the usage pattern and perform its internal optimizations.

https://cloud.google.com/bigtable/docs/performance
https://pixabay.com/photos/chemistry-lab-experiment-3005692/