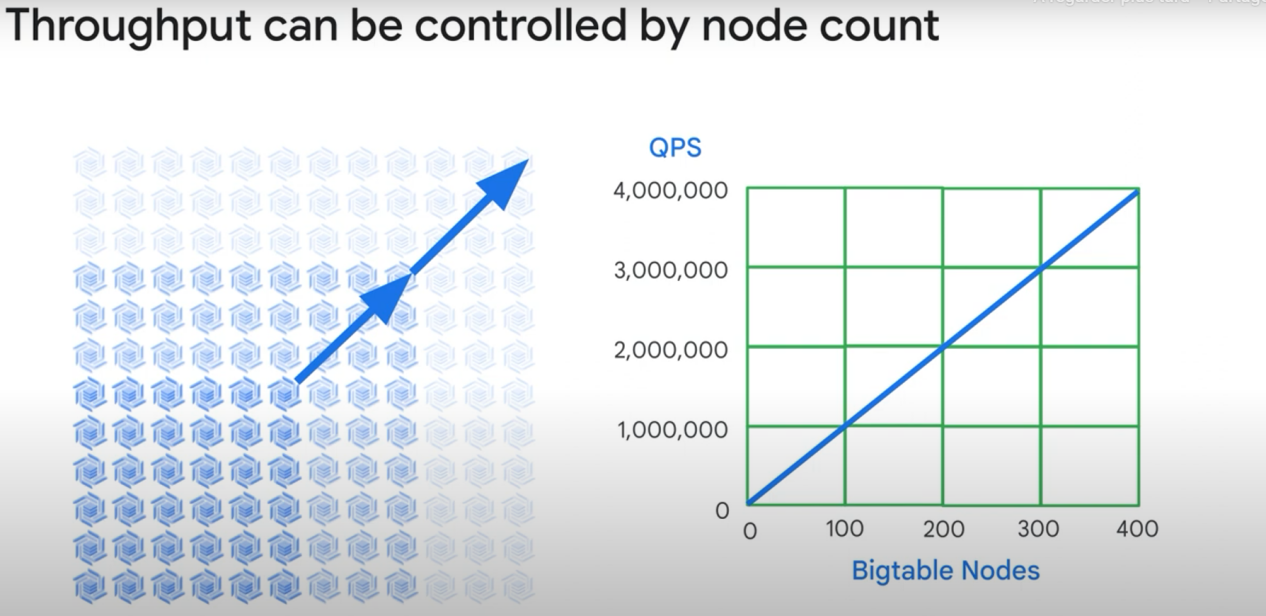


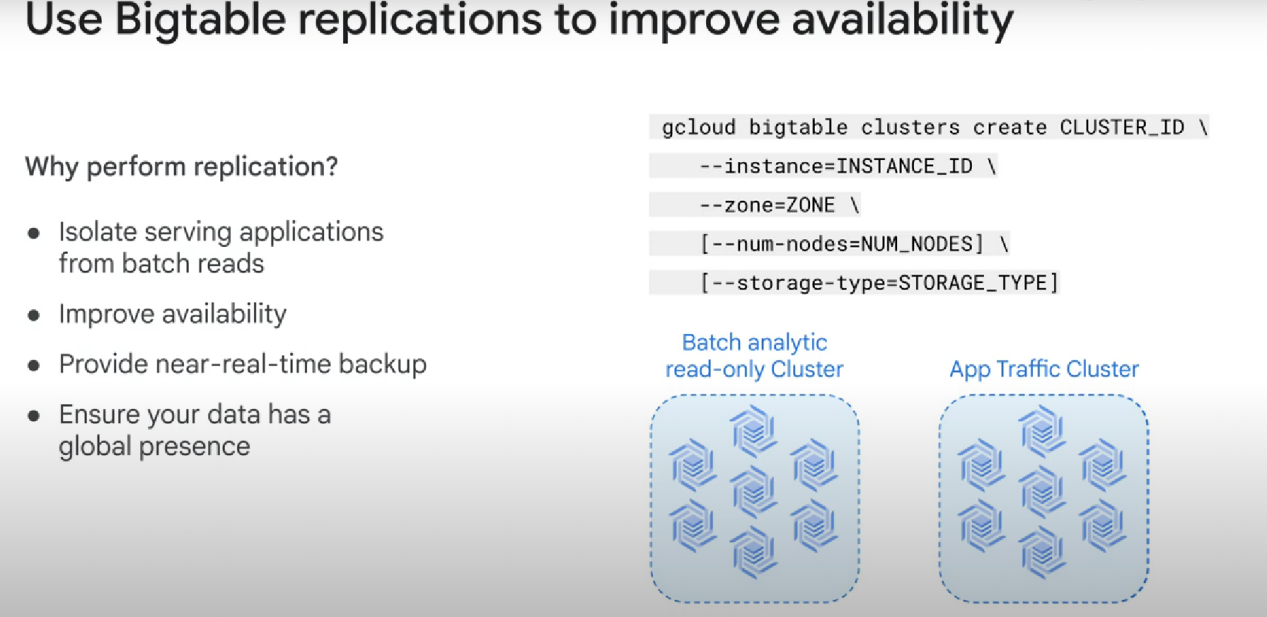
Design your table schema that will allow reads and writes to be evenly distributed across the bigquery cluster.

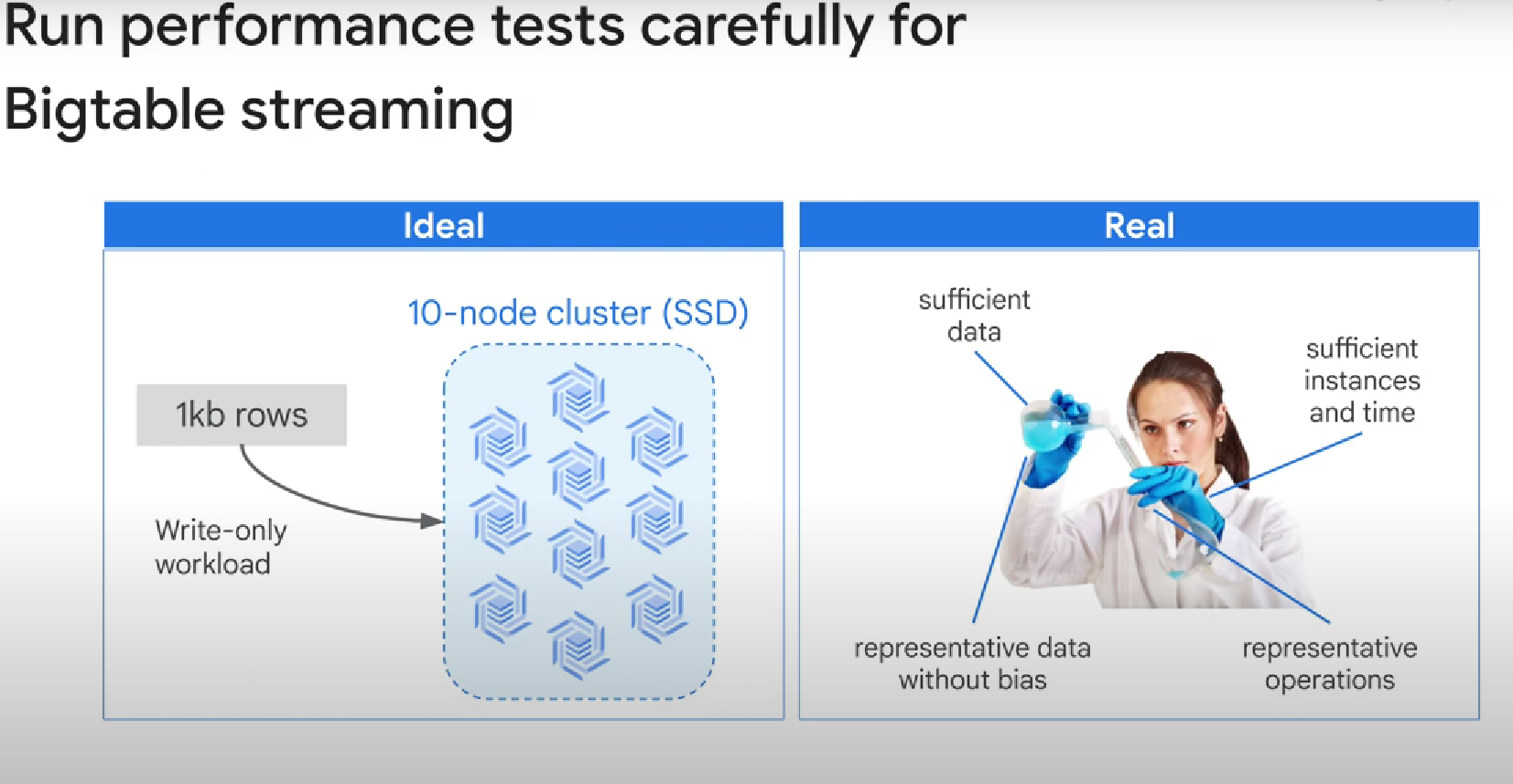


Qps/ QUERY per second

Smaller rows (less columns) = higher throughput

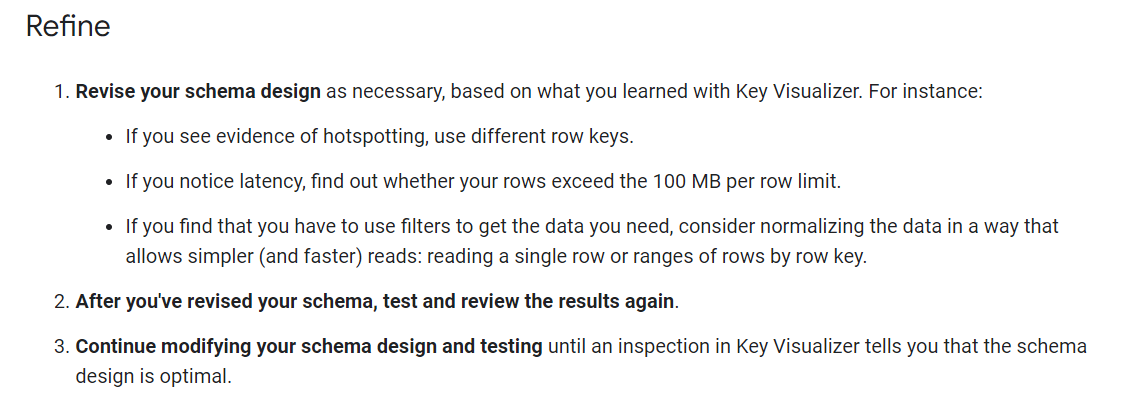
Selecting the right row key is critical. Remember that the rows are stored lexicographically according to the row key. The goal when optimizing for streaming is to avoid creating hot spots when writing, which would cause Bigtable to have to split tablets and adjust loads. To accomplish that, you want the data to be as evenly distributed as possible. Reading delays adding to processing delays leads to response time.

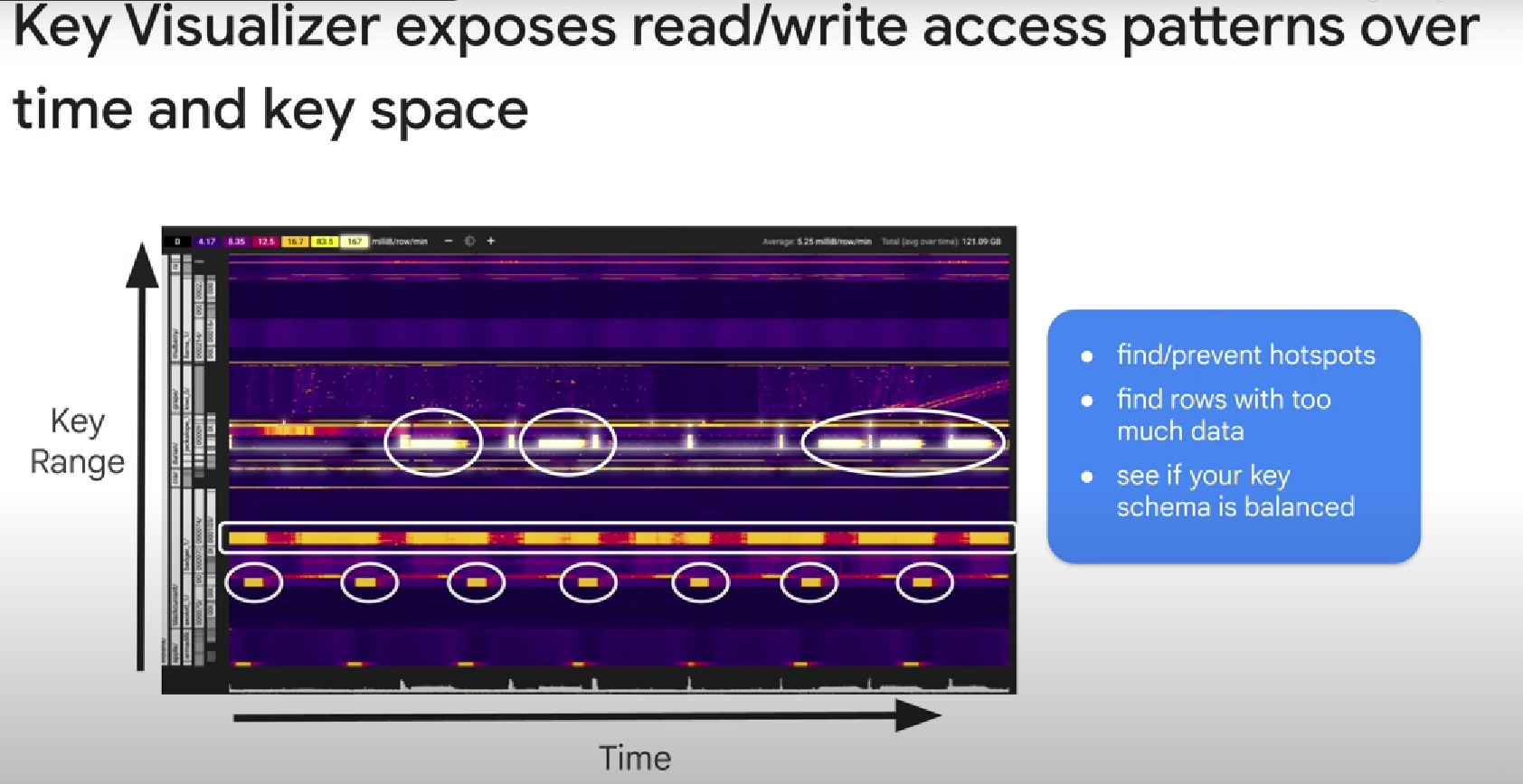




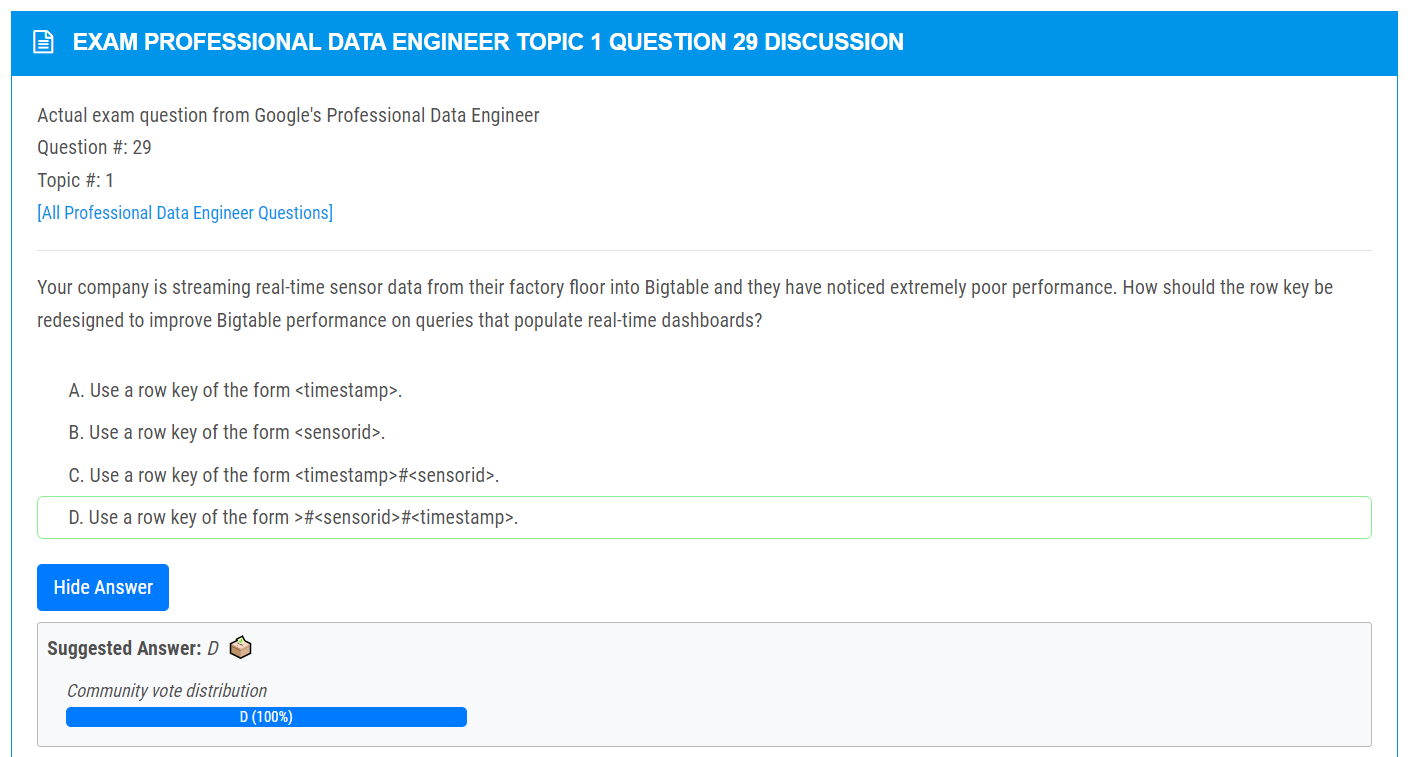
In reality, you need to experiment with the parameters. Run performance tests. Look at the job metrics. RUN THE TESTS ON AT LEAST **300 GB** of data. The test need to last a long enough period of time to give bigtable enough time to optimize itself.

After running the tests:





Key Visualizer is a service.



Each time BigTable looks for data in a table it does a scan and sort operations. By starting each unique id by sensorId it will make it easier to group and sort data since **it has the lowest cardinality**

Note: starting your row key by «timestamp», or using «timestamps» by itself, is a very bad practice

Note 2: adding the «timestamps» to your row key, somewhere in the middle, is good practice!