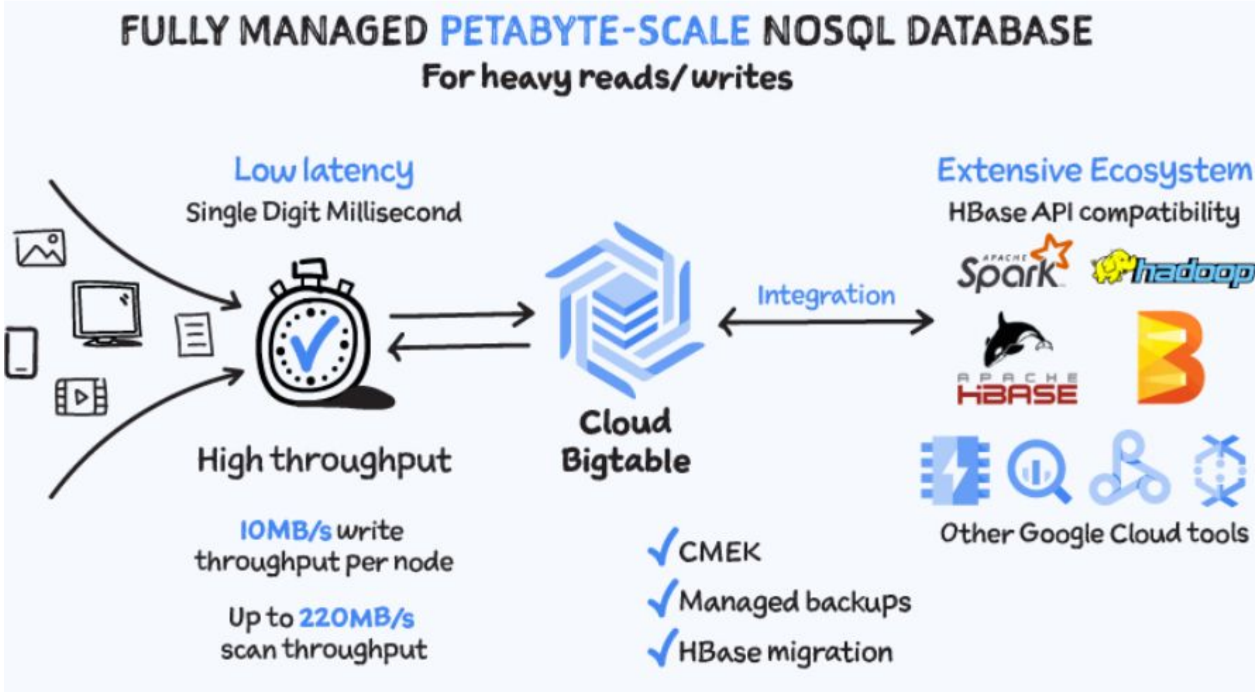
Data Engineering - Final

Google Bigtable

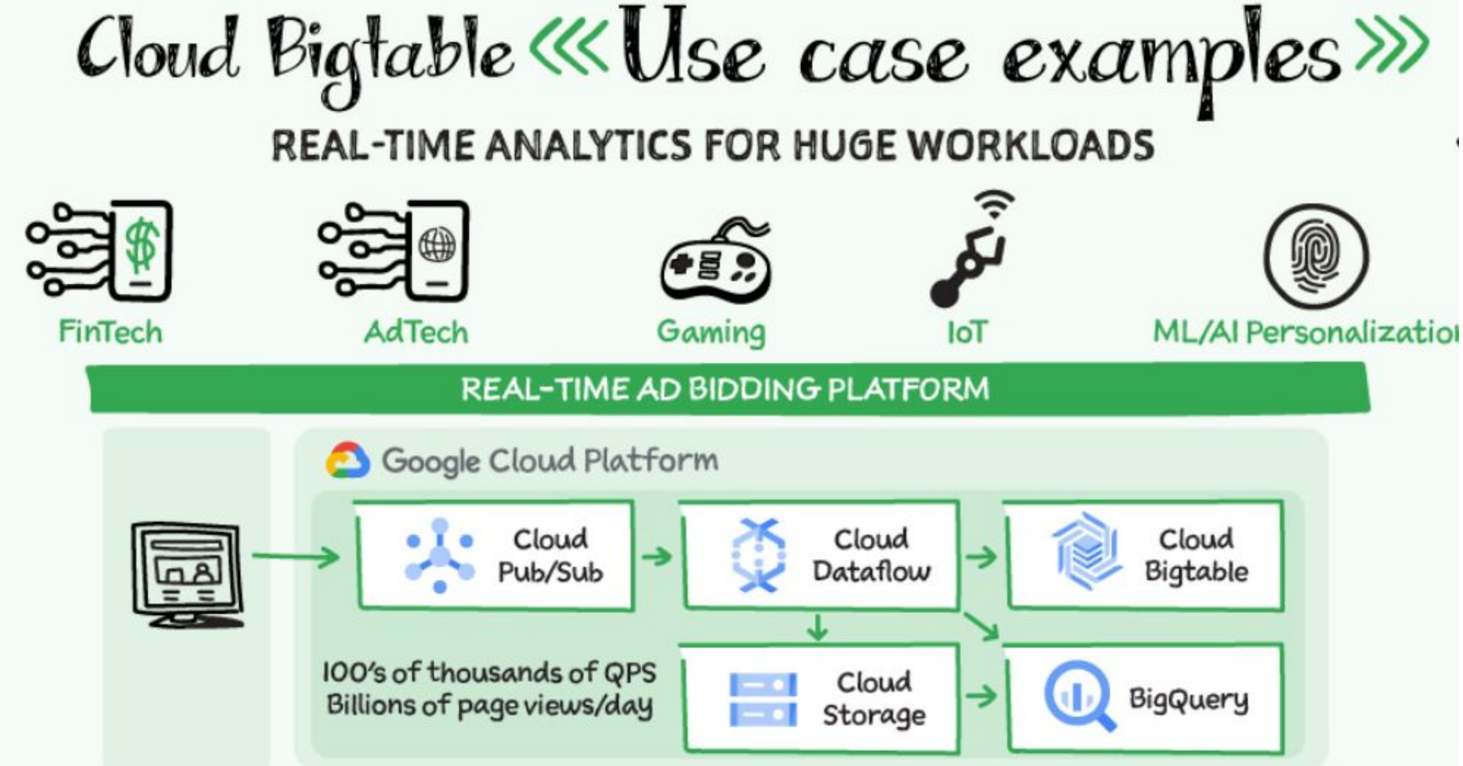
Instructor: Assoc.Prof.Dr. Dang Tran Khanh
Student: 2070682 - Pham Nguyen Nhat Minh

What is Bigtable?



Source: https://thecloudgirl.dev/images/Bigtable.jpg

Bigtable use cases


Cloud Bigtable «Use case examples»
REAL-TIME ANALYTICS FOR HUGE WORKLOADS
FinTech
AdTech
Gaming
IoT
ML/AI Personalizatio
REAL-TIME AD BIDDING PLATFORM
Google Cloud Platform
Cloud Pub/Sub
Cloud Dataflow
Cloud Bigtable
Cloud Storage
BigQuery
100's of thousands of QPS
Billions of page views/day

Source: https://thecloudgirl.dev/images/Bigtable.jpg

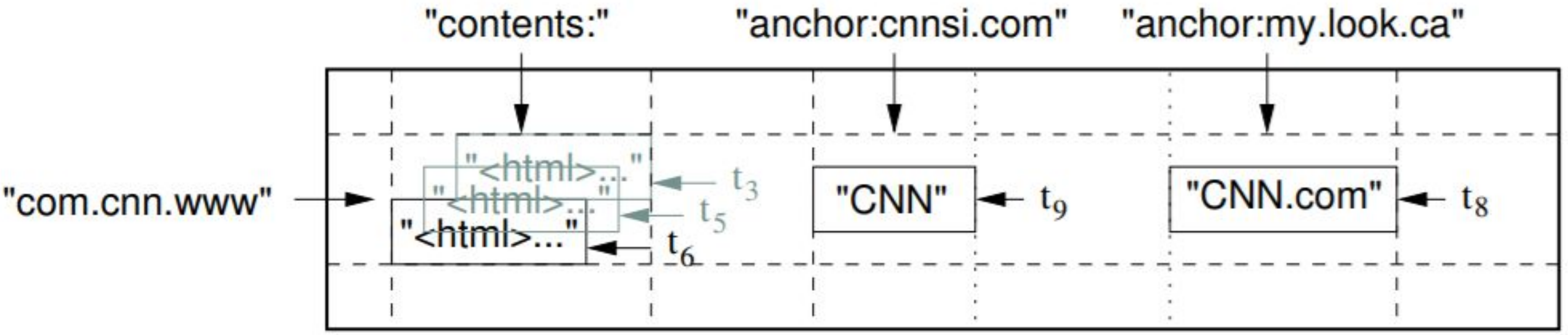
Bigtable data model (key-value based)

- One big map that is indexed by row key, column key and timestamp.

  (row key: String, column key: String, timestamp: Int64) -> index key

- The value is array of bytes that can be interpreted by applications.

Bigtable data model (cont.)



A slice of an example table that stores Web pages

Bigtable characteristics

- The table is **sparse**: different rows in a table may use different columns, with many of the columns empty for a particular row.

- The data is **distributed**: the table is broken up among rows, with groups of adjacent rows managed by a server. But a row itself is never distributed.

- The data is **sorted by keys**: helps keep related data close together, usually on the same machine to optimize the querying speed.

Demo system

Target

- To demonstrate the utility of BigTable in real use case with real dataset.

About the dataset

- Open source data: New York city buses data, with more than 300 routes and 5800 vehicles in one month, the data is nearly 6GB

- Link to dataset: https://www.kaggle.com/stoney71/new-york-city-transport-statistics

Design schema

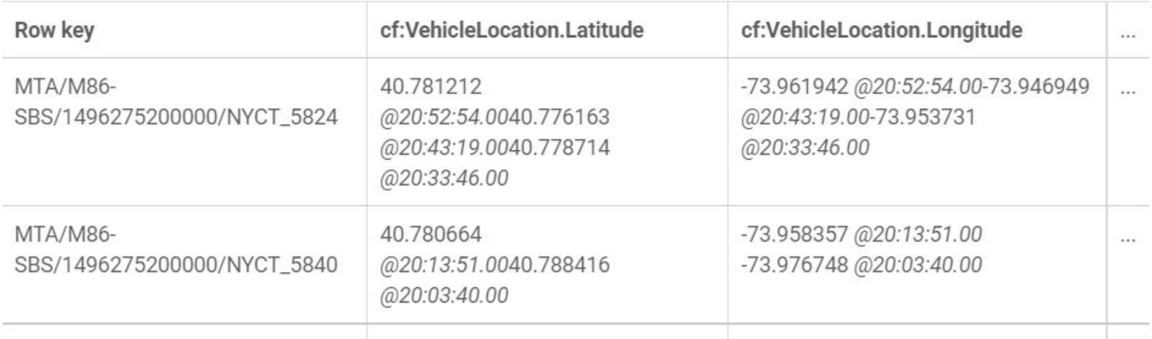
Planning out the query:

- Get the locations of a specific vehicle over an hour.
- Get the locations of an entire bus line over an hour.
- Get the locations of all buses in Manhattan in an hour.

Design the row key with following format: [Bus company/Bus line/Timestamp rounded down to the hour/Vehicle ID]

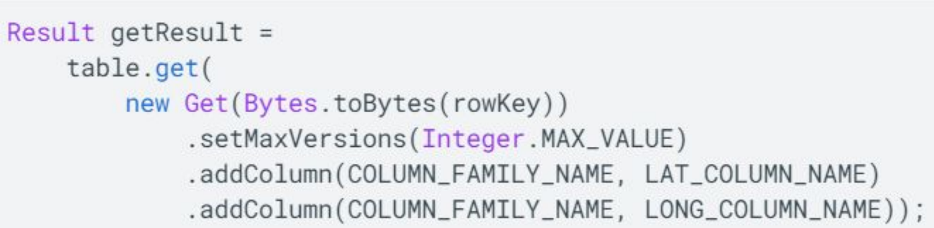
Each row has an hour of data, and each cell holds multiple time-stamped versions of the data.

Design schema (cont.)

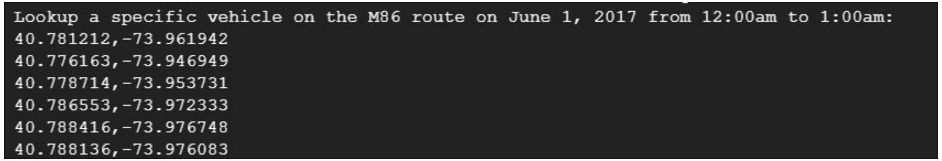


Row key	cf:VehicleLocation.Latitude	cf:VehicleLocation.Longitude	...
MTA/M86-SBS/1496275200000/NYCT_5824	40.781212 *@20:52:54.00*40.776163 *@20:43:19.00*40.778714 *@20:33:46.00*	-73.961942 *@20:52:54.00*-73.946949 *@20:43:19.00*-73.953731 *@20:33:46.00*	...
MTA/M86-SBS/1496275200000/NYCT_5840	40.780664 *@20:13:51.00*40.788416 *@20:03:40.00*	-73.958357 *@20:13:51.00* -73.976748 *@20:03:40.00*	...

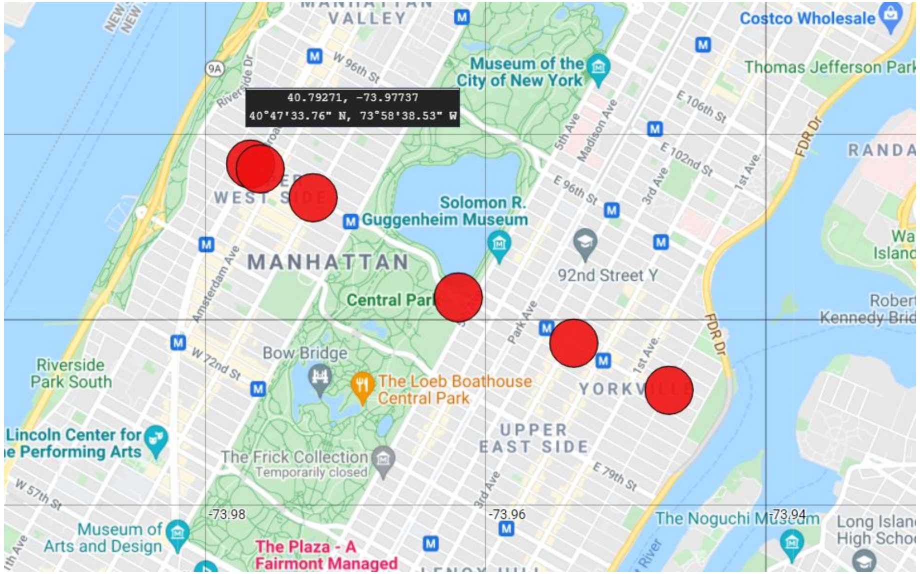
Perform a lookup



```
Result getResult =
    table.get(
        new Get(Bytes.toBytes(rowKey))
            .setMaxVersions(Integer.MAX_VALUE)
            .addColumn(COLUMN_FAMILY_NAME, LAT_COLUMN_NAME)
            .addColumn(COLUMN_FAMILY_NAME, LONG_COLUMN_NAME));
```



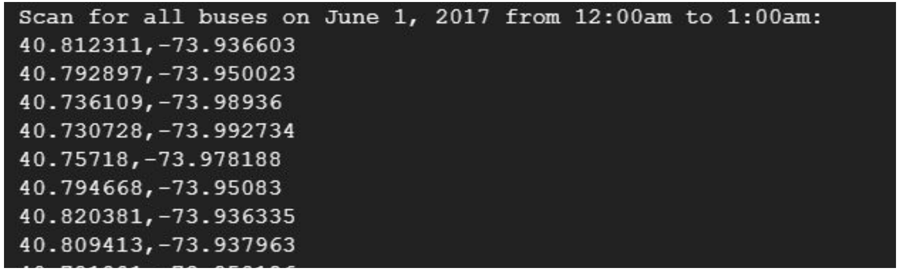
```
Lookup a specific vehicle on the M86 route on June 1, 2017 from 12:00am to 1:00am:
40.781212,-73.961942
40.776163,-73.946949
40.778714,-73.953731
40.786553,-73.972333
40.788416,-73.976748
40.788136,-73.976083
```



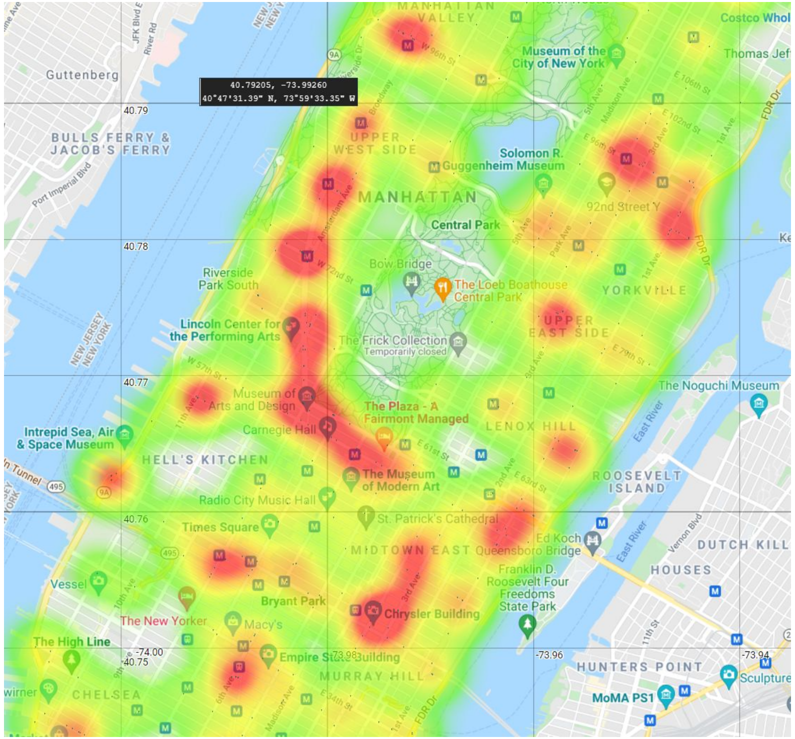
Perform a scan

```
private static final String[] MANHATTAN_BUS_LINES = {"M1","M2","M3",...

Scan scan;
ResultScanner scanner;
List<RowRange> ranges = new ArrayList<>();
for (String busLine : MANHATTAN_BUS_LINES) {
  ranges.add(
      new RowRange(
          Bytes.toBytes("MTA/" + busLine + "/1496275200000"), true,
          Bytes.toBytes("MTA/" + busLine + "/1496275200001"), false));
}
Filter filter = new MultiRowRangeFilter(ranges);
scan = new Scan();
scan.setFilter(filter);
scan.setMaxVersions(Integer.MAX_VALUE)
    .addColumn(COLUMN_FAMILY_NAME, LAT_COLUMN_NAME)
    .addColumn(COLUMN_FAMILY_NAME, LONG_COLUMN_NAME);
scan.withStartRow(Bytes.toBytes("MTA/M")).setRowPrefixFilter(Bytes.toBytes("MTA/M"));
scanner = table.getScanner(scan);
```



```
Scan for all buses on June 1, 2017 from 12:00am to 1:00am:
40.812311,-73.936603
40.792897,-73.950023
40.736109,-73.98936
40.730728,-73.992734
40.75718,-73.978188
40.794668,-73.95083
40.820381,-73.936335
40.809413,-73.937963
```



Summary

- Research on the technical design of the Google Bigtable technology
- Implement a demo system on real dataset using Google Bigtable technology

Thank you!!!