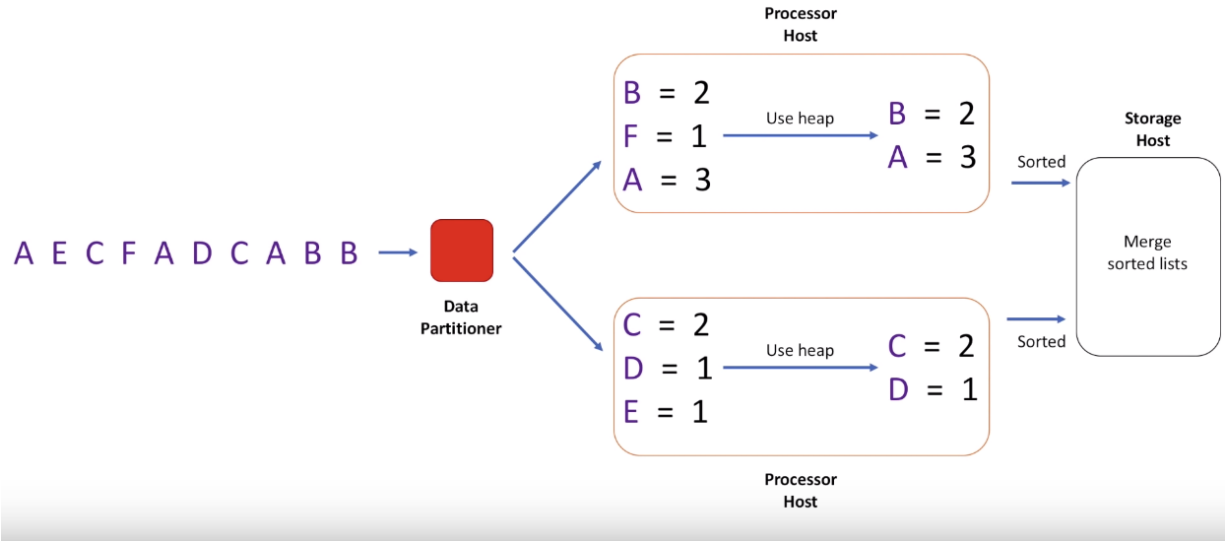
Top K Frequent Hitters

Return a list of most viewed videos for the last several minutes
Stream processing problem

topK(k, startTime, endTime)

Single host:
Load into memory
Hash table —> sort or heap
Leetcode TopK algorithm

min heap: size K



Processor hosts only pass a list of size k to the storage hosts
We don't pass the whole hash table

Q: streaming data is unbounded, infinite
Storage host stores a list of heavy hitters for **every minute**

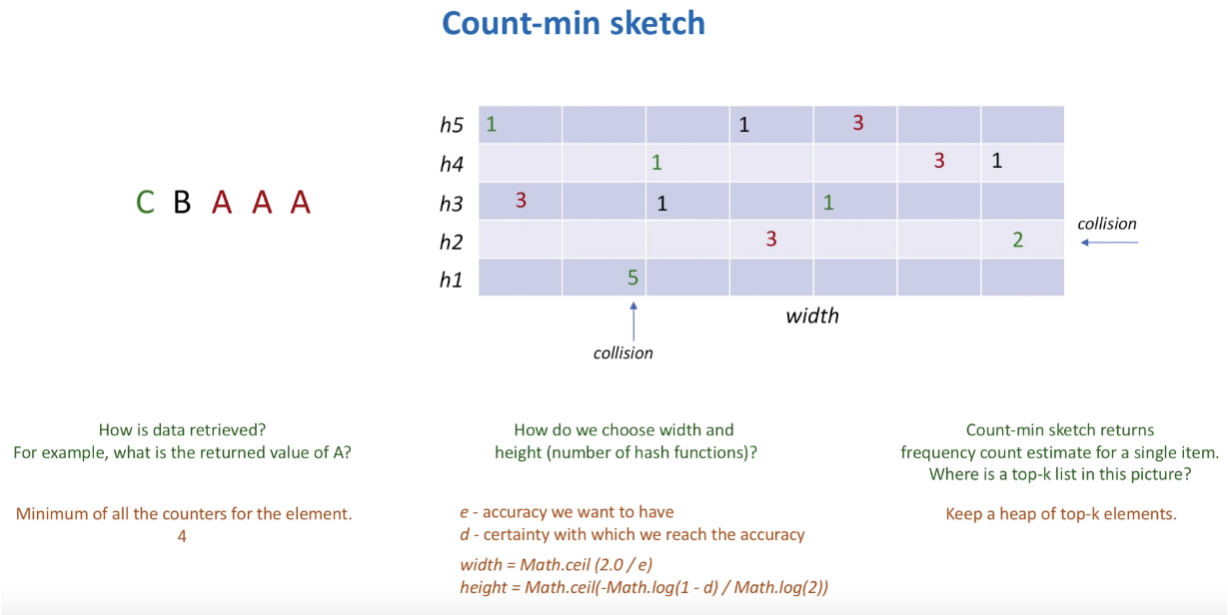
Q: what if we want to know heavy hitters for 1-hour or 1-day period?

Need the data for 1 day
Store all the data in disk and use batch processing framework to calculate
MapReduce

Data Partitioning: data replication
Each partition are stored in different nodes
Rebalancing: when a new node is added to the cluster or removed

Simpler solution:
Using fixed size memory but the result is not 100% accurate

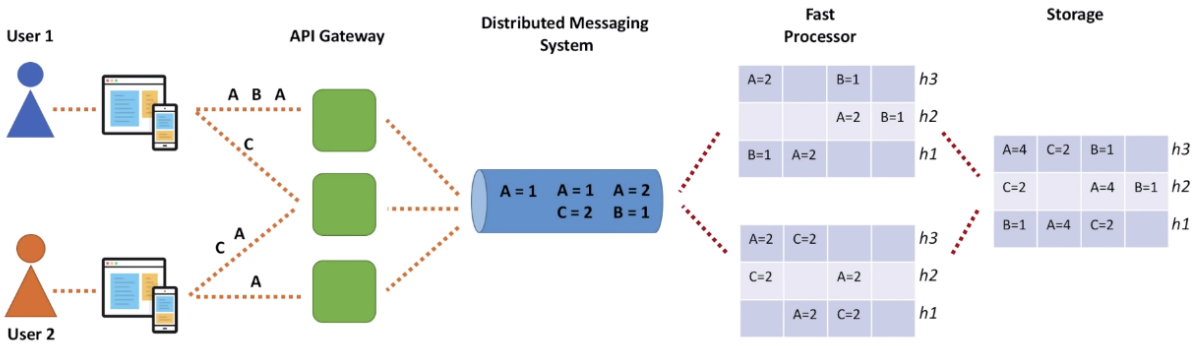
Count-min Sketch
Replace the hash table which can grow big with a count-min-sketch that always has predefined size



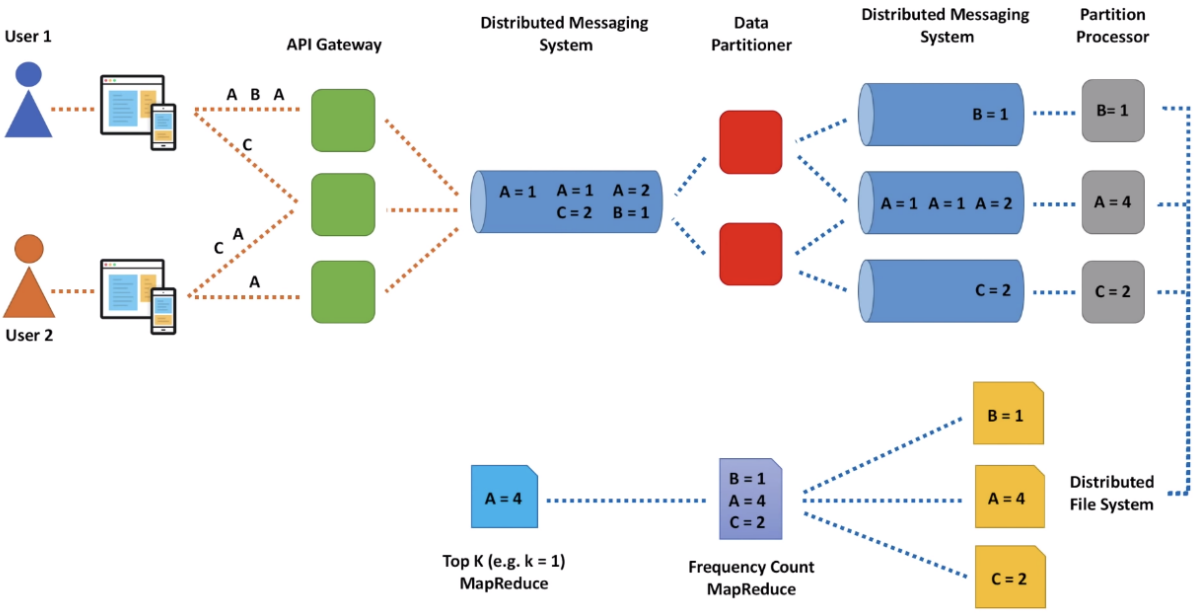
High Level Design
Fast Path: count min sketch
Slow Path: HDFS, MapReduce
2 MapReduce Jobs: Frequency Count MapReduce Job, Top K MapReduce Job
Something faster than MapReduce: partition again

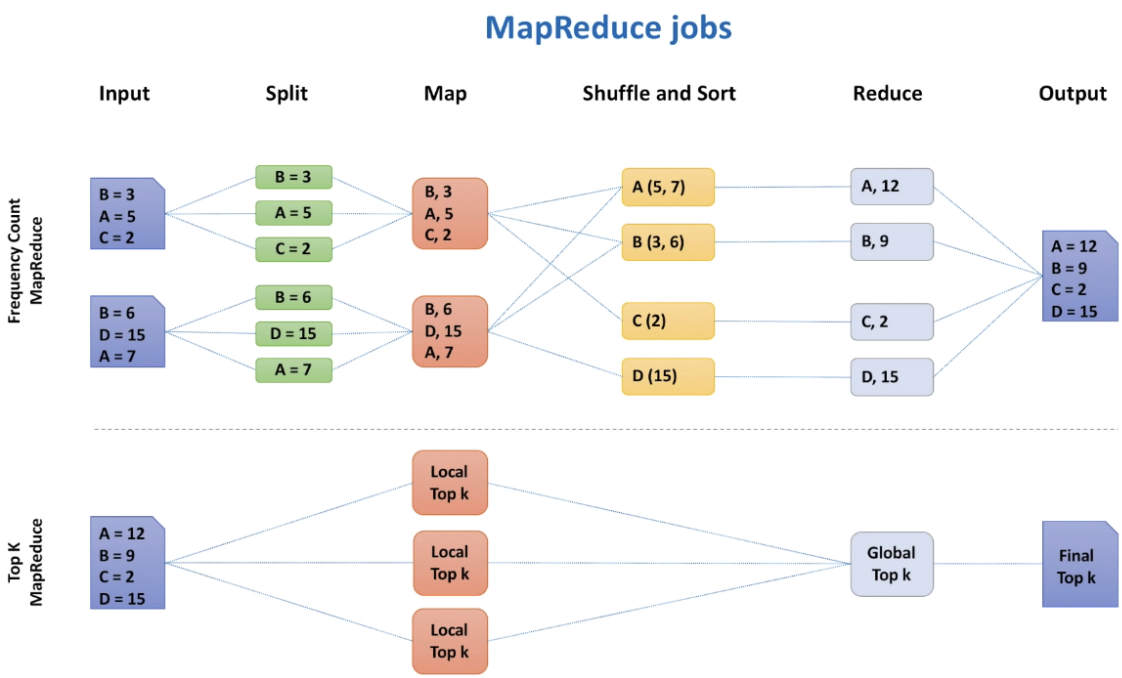
API gateway hosts: log the requests; hash table: count of views (# times viewed / video in last several seconds) aggregated on each host
Information about the views are sent to the distributed messaging system

Data flow, fast path



Data flow, slow path





Data Retriever: