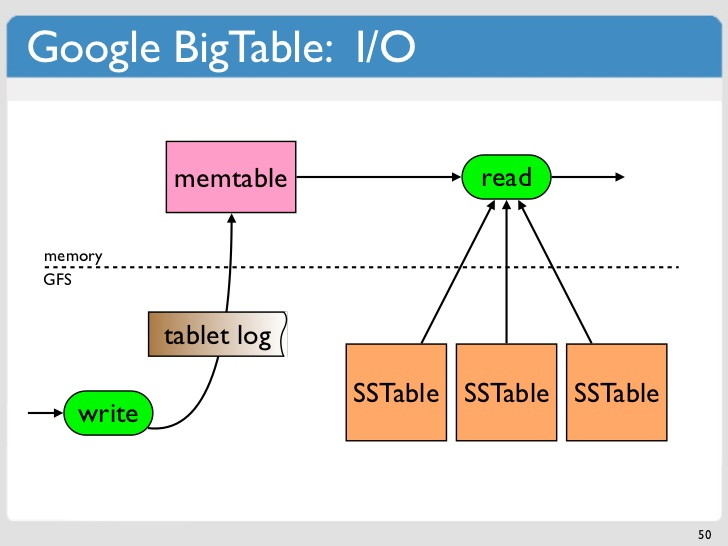
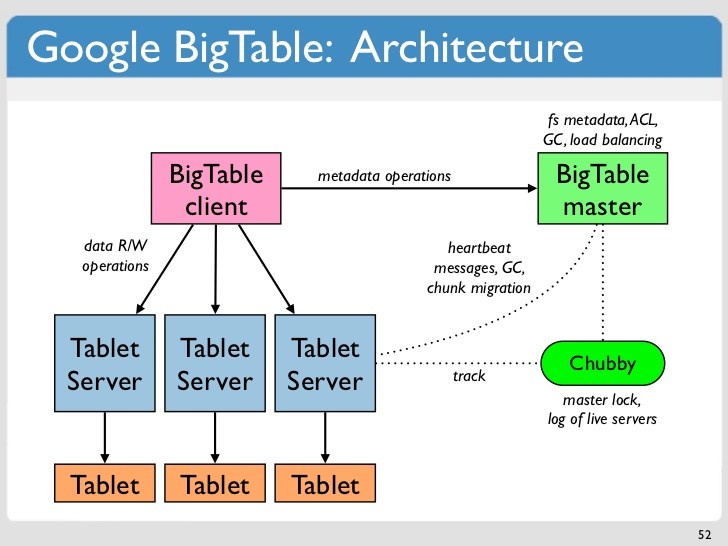
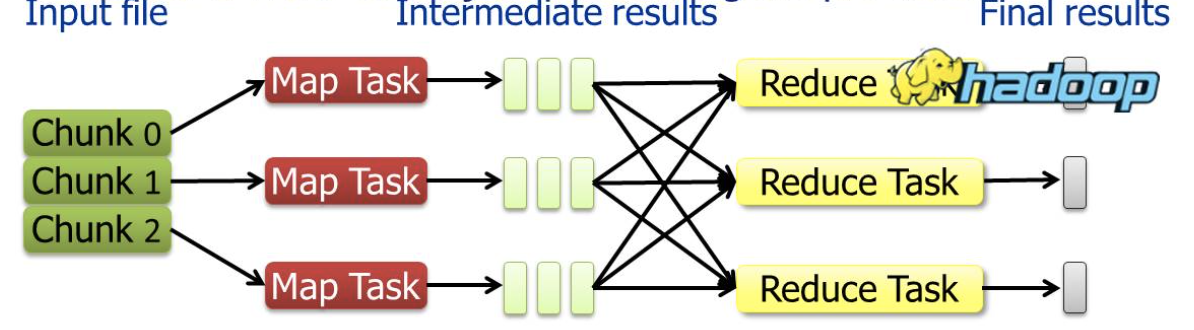
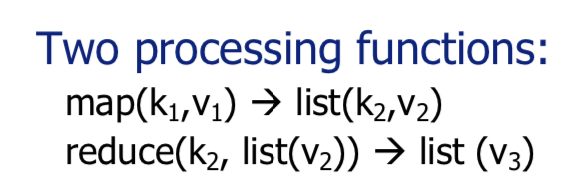
1. 1. 1. Soft lease – when lease expires, the provider holds onto the user’s cache until and unless they need to hand out that address/space or they receive instructions on how to clear their cache
      2. Coordination service – Service which provides coordination for processes sent to it and ensures the distributed system acts accordingly
      3. Data locality – essentially how close the data is to the computation; moving computation closer to the data saves bandwidth => Tasks are executed on the node containing the data
      4. Tail latency – small percentage of response times that take the longest in comparison to the bulk of its response times
   2. 1. 



* + 1. Each read/write operation done on a row is atomic, but there is no atomicity across rows. This was done to simplify concurrency controls, such that the user can do multiple row updates on multiple servers without any coordination required.
    2. Group together rows by column families such that each tablet is responsible for a column family and keep a data structure like memtables to record any operations that ought to take place on every row. Include an API calls for multi-row transactions e.g., INSERT, UPDATE
    3. Updating a column value on an already existing column that only occurs within a column family
    4. Inserting a new column and value into many rows across different column families.

1. 1. 1. Quorum – A quorum is the minimum number of votes that a distributed transaction must obtain in order to be allowed to perform an operation in a distributed system. A quorum-based technique is implemented to enforce consistent operation in a distributed system.
      2. Distributed hash table – A distributed hash table is a distributed system that provides a lookup service like a hash table: key-value pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key.
      3. Vector Clocks are used in a distributed system to determine whether pairs of events are causally related. Using Vector Clocks, timestamps are generated for each event in the system, and their causal relationship is determined by comparing those timestamps.
      4. A distributed transaction is a set of operations on data that is performed across two or more data repositories (especially databases). It is typically coordinated across separate nodes connected by a network but may also span multiple databases on a single server.
   2. 1.  We first partition our input file into chunks, where each chunk is independently processed by a map task and produces intermediate results. These results are then shuffled (distributed across various reduce tasks depending on some partitioning) which the reduce task aggregates these results and formulate the final desired result.
      2. 
      3. We can process the required query with a map task over each individual dataset in order to obtain the intermediate results for each dataset, which we can combine using a single reduce task to achieve the overall result. This is essentially the same as using JOINs over multiple datasets and calling the required query.
2. 1. Since MapReduce produces intermediate results, with billions of users it would have to iterate through and consequently produce lots of intermediate results, which become very expensive and very inefficient for this scenario.
   2. We can represent each node as a separate row with the user’s name as the row index and introduce the user’s friendships (i.e., edges connected to that node) as columns with the column name being their friend’s name (I.e., their row index) and the value being 1. This allows for each user to be easily represented as a row and to have few too many columns depending on how many friends they have. We can have column families set up for each of the top influencers.
   3. {{Insert architecture of BigTable}}
   4. R1: BigTable is designed around supporting billions of rows and columns, so this requirement is fulfilled.  
      R2: BigTable is built around being scalable, and it’s easy to insert new rows for new users and to insert new columns for new friendships formed.  
      R3: BigTable is designed to be sparsely filled so this requirement is also fulfilled.