BigTable & Cassandra

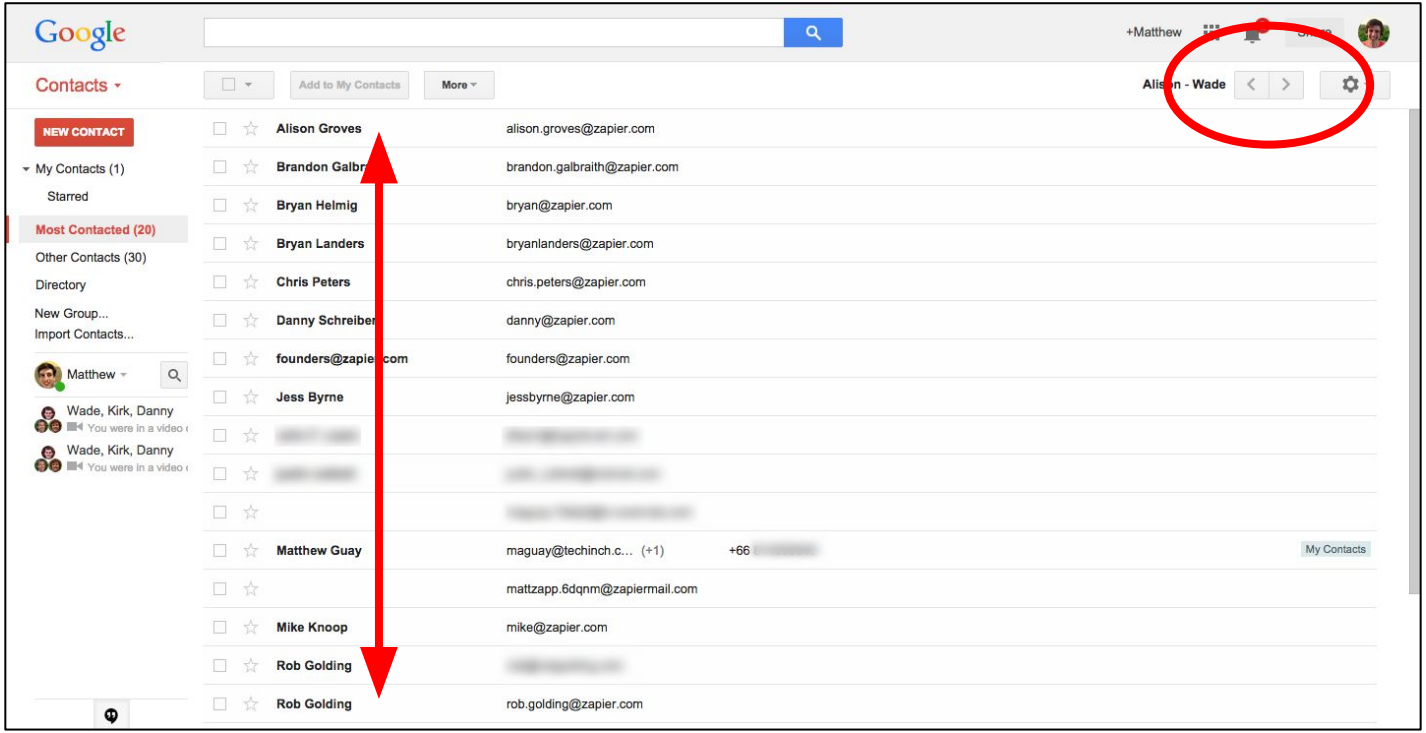
Ed Schouten ‹ed@nuxi.nl›
Ron Lievens ‹ron.lievens@gmail.com›

Problem definition

How do you make a fast, scalable database for at least these queries?

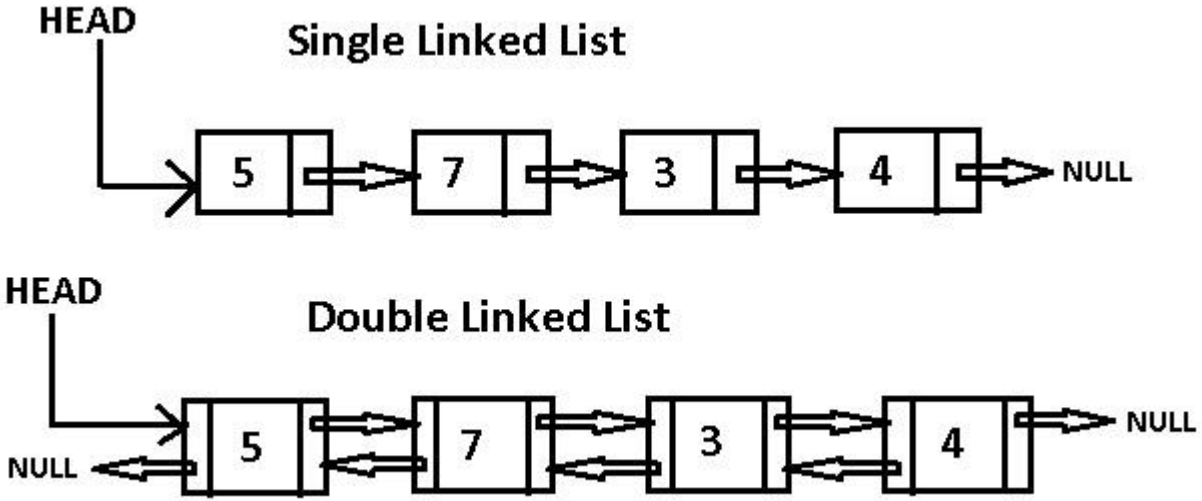
- `Insert : K, V`
- `Get : K ➜ V`
- `Delete : K`
- `Scan : K, n ➜ K[0 … n]`

What is Scan?

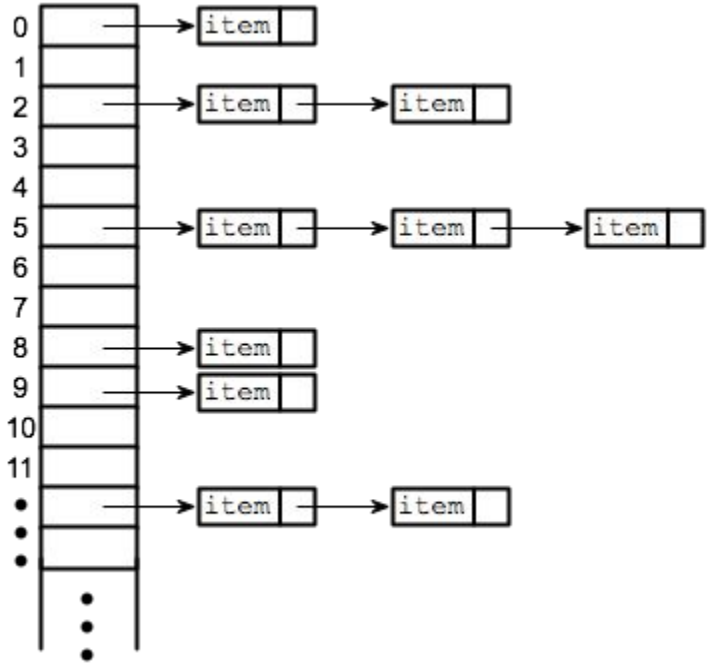

Google
+Matthew
Contacts
Add to My Contacts
More
Alison - Wade
NEW CONTACT
My Contacts (1)
Starred
Most Contacted (20)
Other Contacts (30)
Directory
New Group...
Import Contacts...
Matthew
Wade, Kirk, Danny
Wade, Kirk, Danny
Alison Groves alison.groves@zapier.com
Brandon Galbr brandon.galbraith@zapier.com
Bryan Helmig bryan@zapier.com
Bryan Landers bryanlanders@zapier.com
Chris Peters chris.peters@zapier.com
Danny Schreiber danny@zapier.com
founders@zapie com founders@zapier.com
Jess Byrne jessbyrne@zapier.com
Matthew Guay maguay@techinch.c... (+1) +66
My Contacts
mattzapp.6dqnm@zapiermail.com
Mike Knoop mike@zapier.com
Rob Golding
Rob Golding rob.golding@zapier.com

Database in memory

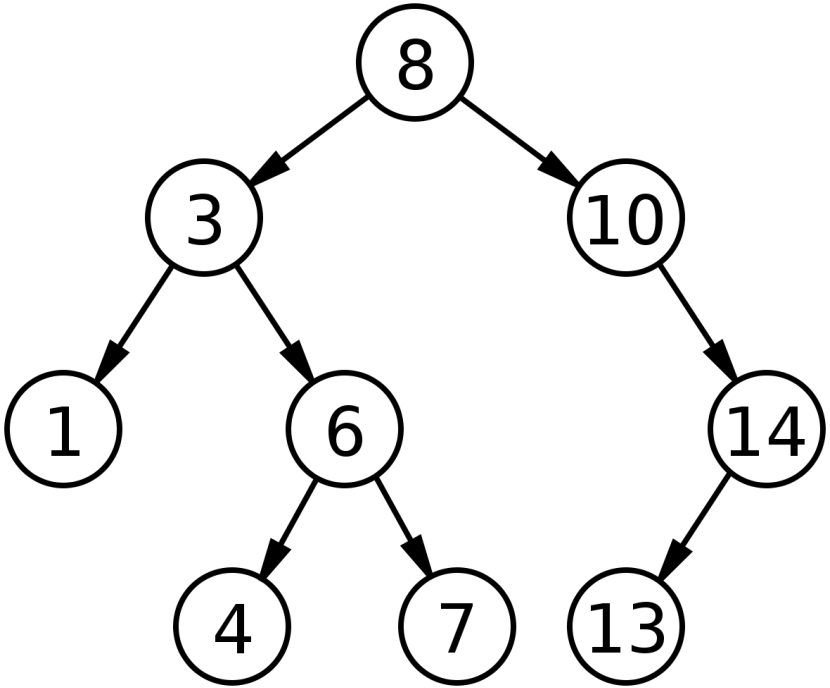
Solution 1: linked list?



Solution 2: a hash table?



Solution 3: a binary search tree?



Performance

Operation	Linked list	Hash table	Binary search tree
Insert	O(n)	O(1)	O(log n)
Get	O(n)	O(1)	O(log n)
Delete	O(n)	O(1)	O(log n)
Scan	O(n + ...)	O(n + ...)	O(log n + k)

Database on disk

SSTables: Sorted String Tables

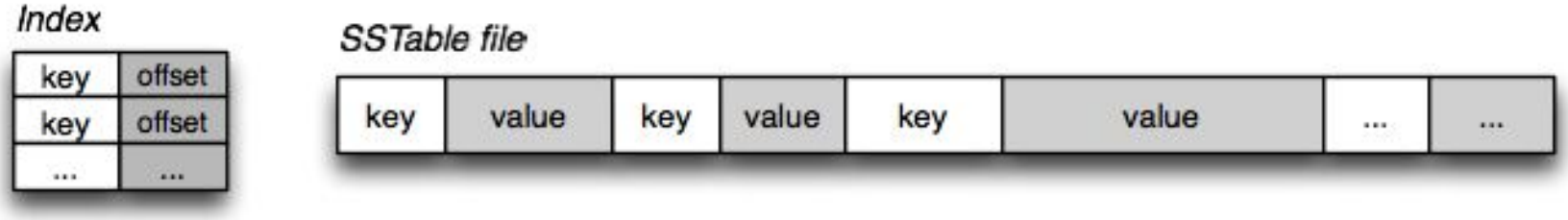
Simple file format:

- Key #1
- Value #1
- Key #2
- Value #2
- ...

Sorted, meaning that entries can be looked up through binary searching.

Making SSTables faster

- Speeding up negative lookups: bloom filters.
  - Bit mask to quickly test whether an element is **certainly not** in the SSTable.
  - Not suited to test whether an element certainly is in the SSTable.
  - “[...], fewer than 10 bits per element are required for a 1% false positive probability, [...]”
- Binary search on SSTables with variable sized elements is hard.
  - Solution: separate index files to store offsets of entries in the data file.



Performance

	MemTables			
Operation	**Linked list**	**Hash table**	**Binary search tree**	**SSTables**
Insert	O(n)	O(1)	O(log n)	?
Get	O(n)	O(1)	O(log n)	O(log n)
Delete	O(n)	O(1)	O(log n)	?
Scan	O(n + ...)	O(n + ...)	O(log n + k)	O(log n + k)

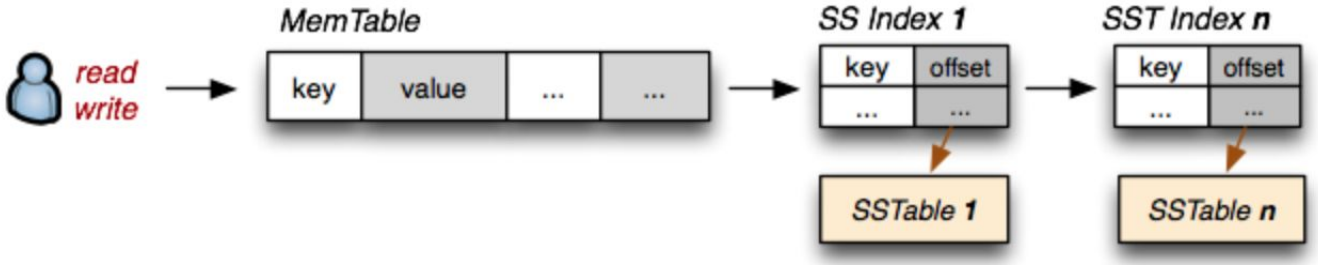
How can we do inserts/deletes on SSTables?

- SSTables are sorted.
  - Adding an element to the start requires a full rewrite.
- Bloom filters don’t allow deletion.
  - Deleting an element would require full recomputation.
- **In short: practically impossible.**

Tablets / Log Structured Merge-trees

What if we combine MemTables and SSTables?

1. Start off with an empty MemTable.
2. Fill the MemTable with records.
3. Out of memory? Write the MemTable to disk as an SSTable.
4. Create a new MemTable.
5. Fill the MemTable with differences w.r.t. the SSTable.
6. Out of memory? Write the MemTable to disk as a second SSTable.



LSM FAQ

- How do you update existing records?
  - You save them as new records in the MemTable.
- How do you delete existing records
  - You insert tombstone records: placeholders to hide the old entry.
- Can this be fast if you have a thousand SSTables?
  - No.
  - Periodic merge compactions recombine old SSTables.
- Implementations?
  - LevelDB by Google: http://leveldb.org/

BigTable

A tablet distribution system

- Start with a single tablet on a single server.
- Is the tablet becoming too large (>128 MB)? If so:
  - Cut the tablet into two smaller ones: ‘tablet splitting’.
  - Optional: migrate one half to another server.
- ‘Hot tablets’ (due to high load) may also cause tablet splitting.
  - Extreme case: single row, having its own tablet and server.
  - Pre-splitting: forced splitting to prepare for future traffic spikes.

Multiple columns in one BigTable?

- Solution 1: store structs/tuples as a value.
  - Google Protobuf and Apache Thrift are good for this.
  - Advantage: consistent updates across the entire row.
  - Disadvantage: `Get()` will always return all of the data.
- Solution 2: column families.
  - Actual separate columns in the BigTable.
  - Every column family is stored by its own set of tablets.
  - Useful when one table has multiple consumers.
  - Also allows for sparse rows.
- In practice, both solutions are combined.

Example of column families

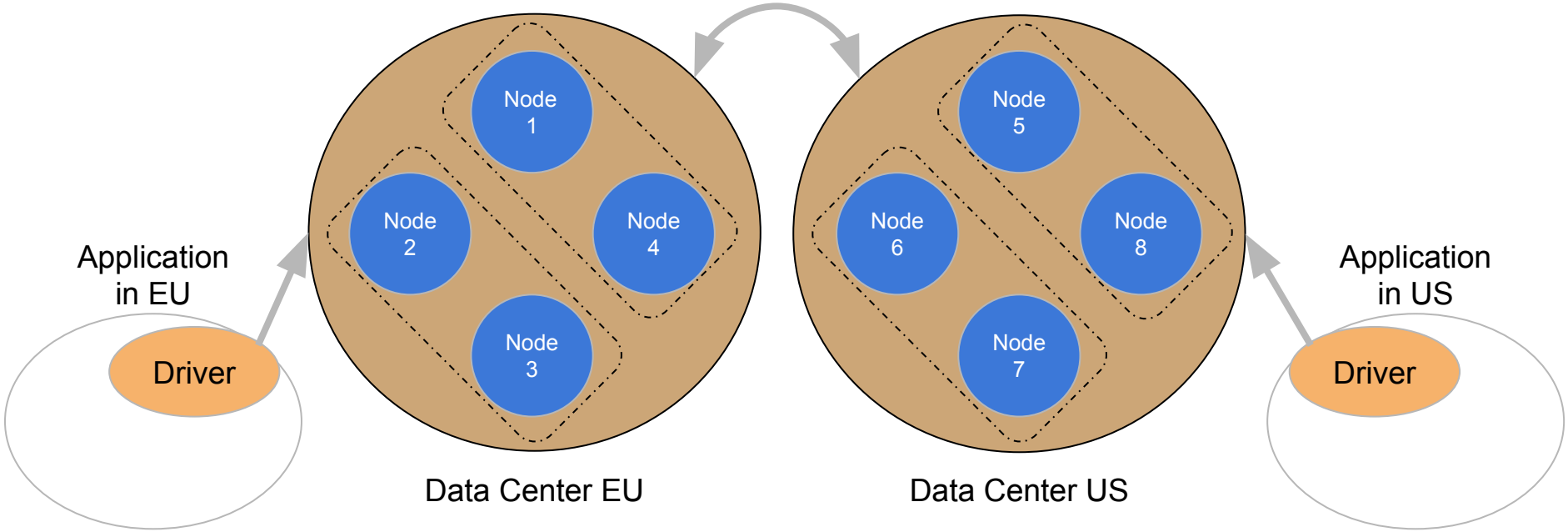
Keys	Address (CF)				Phone numbers (CF)		
	Street	House #	Postal code	City	Home	Office	Mobile
Arnold Schwarzenegger							
Bruce Willis							
Chuck Norris							
Jason Statham							
Jean-Claude Van Damme							
Jet Li							
Sylvester Stallone							
Terry Crews							

Apache Cassandra

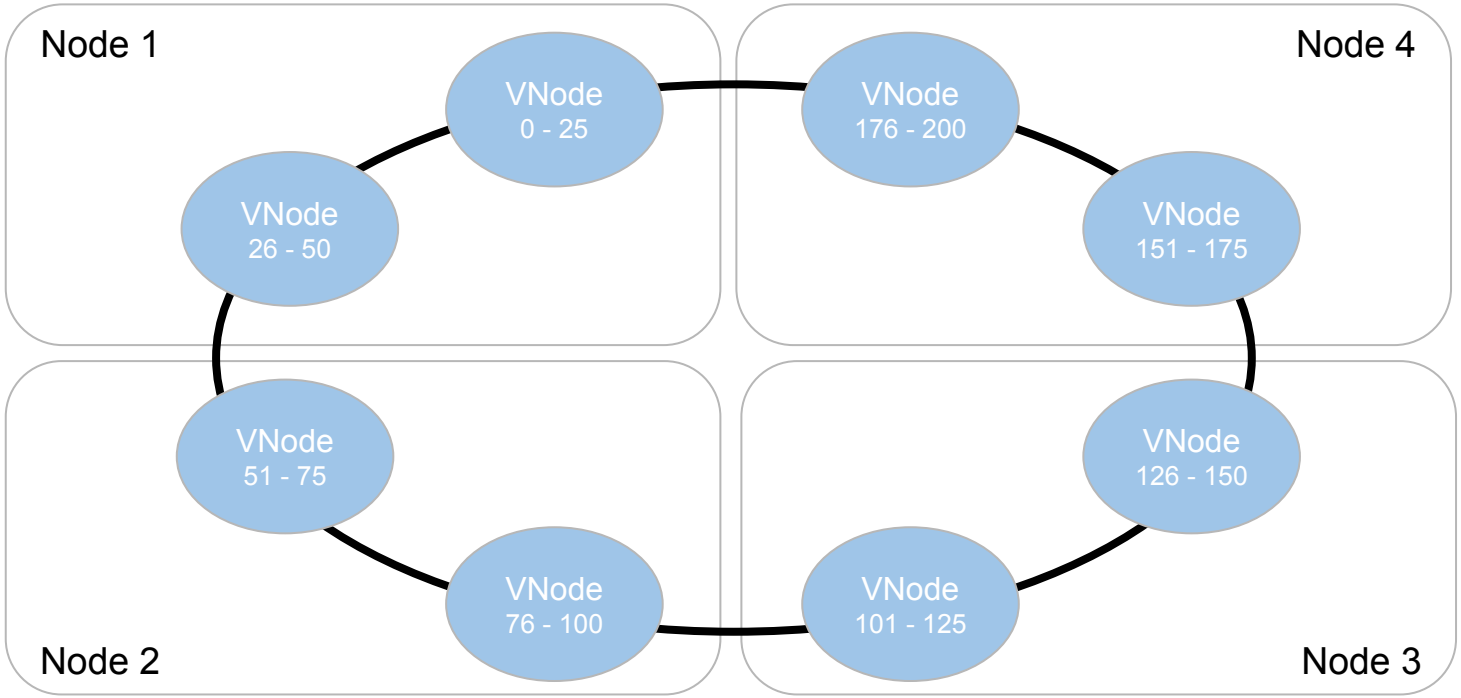
Data structure

BigTable	Cassandra	PostgreSQL
Table	Key space	Schema
Column family	Table	Table
Row	Row	Row
-	Column	Column
Tablet	Partitioning	Partitioning

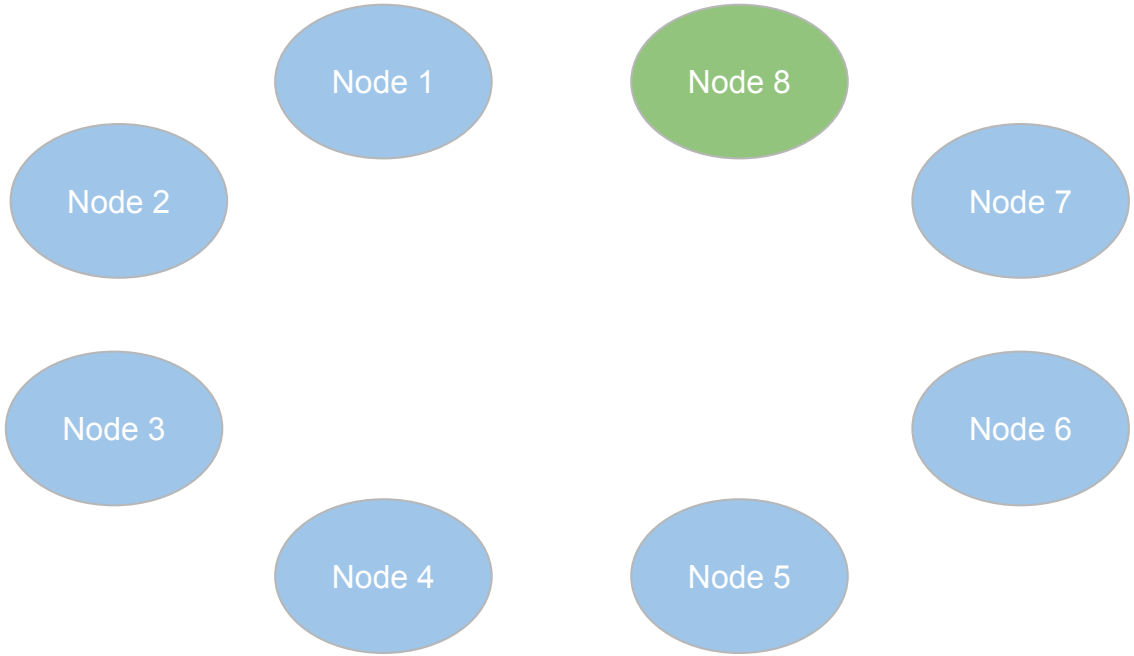
Cassandra overview



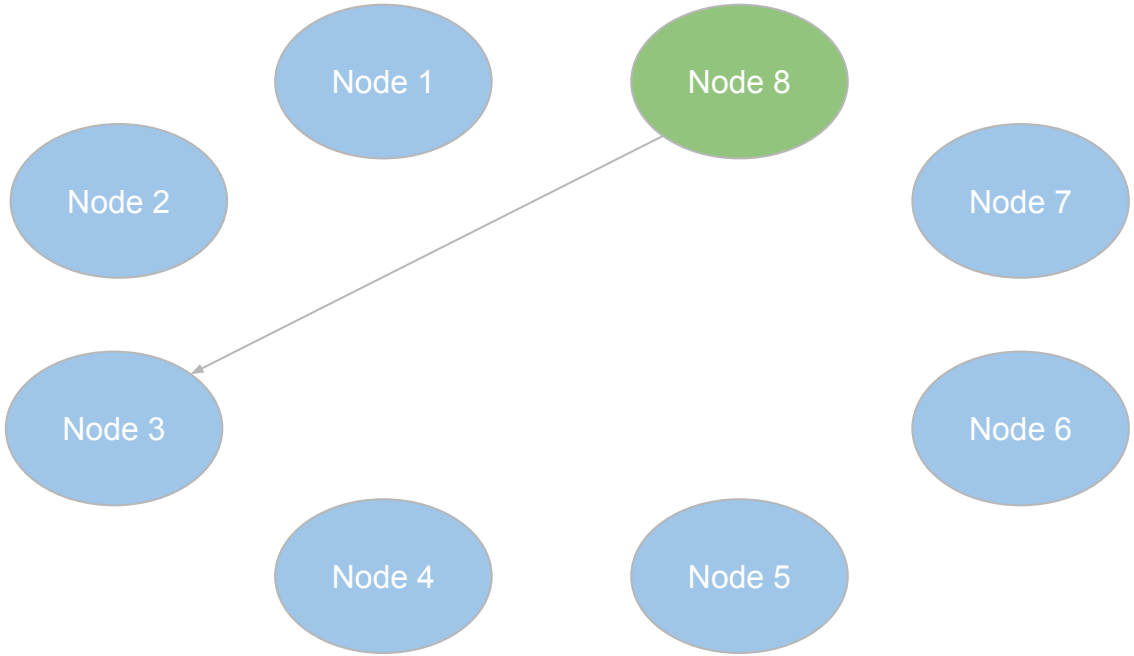
Distributed (murmur3) hash table



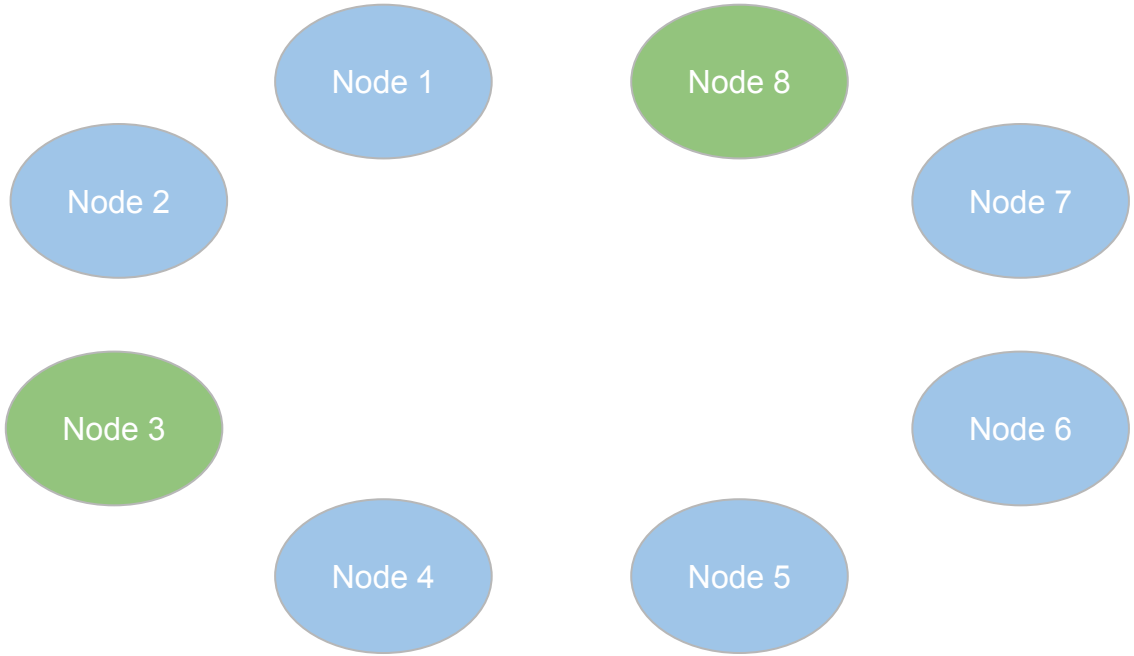
Gossip



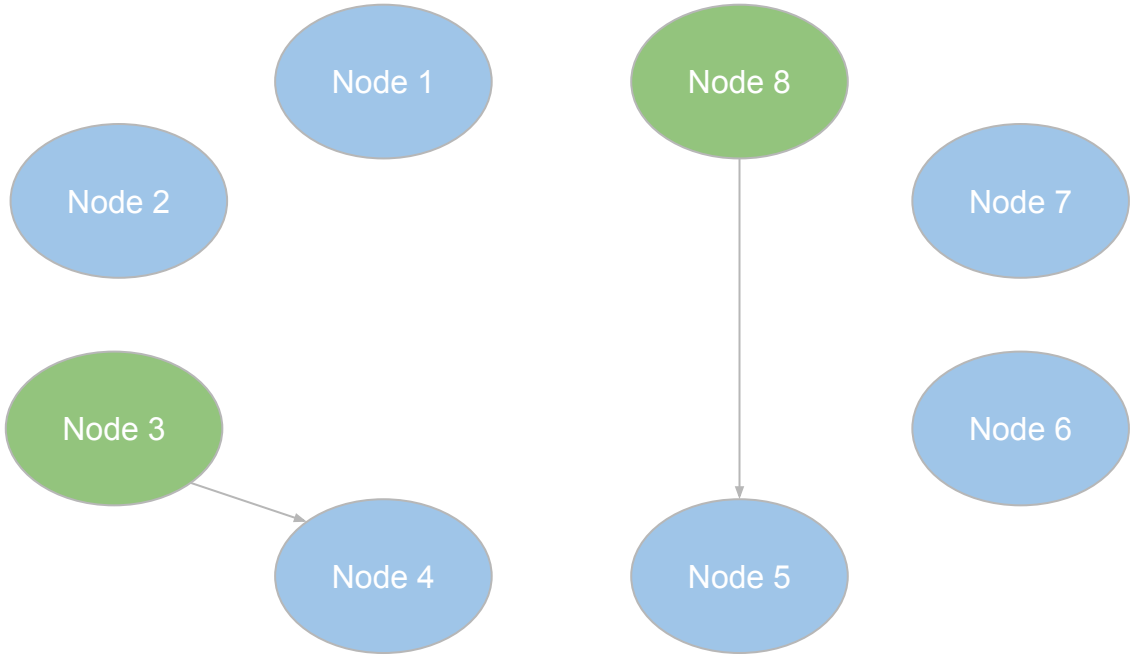
Gossip



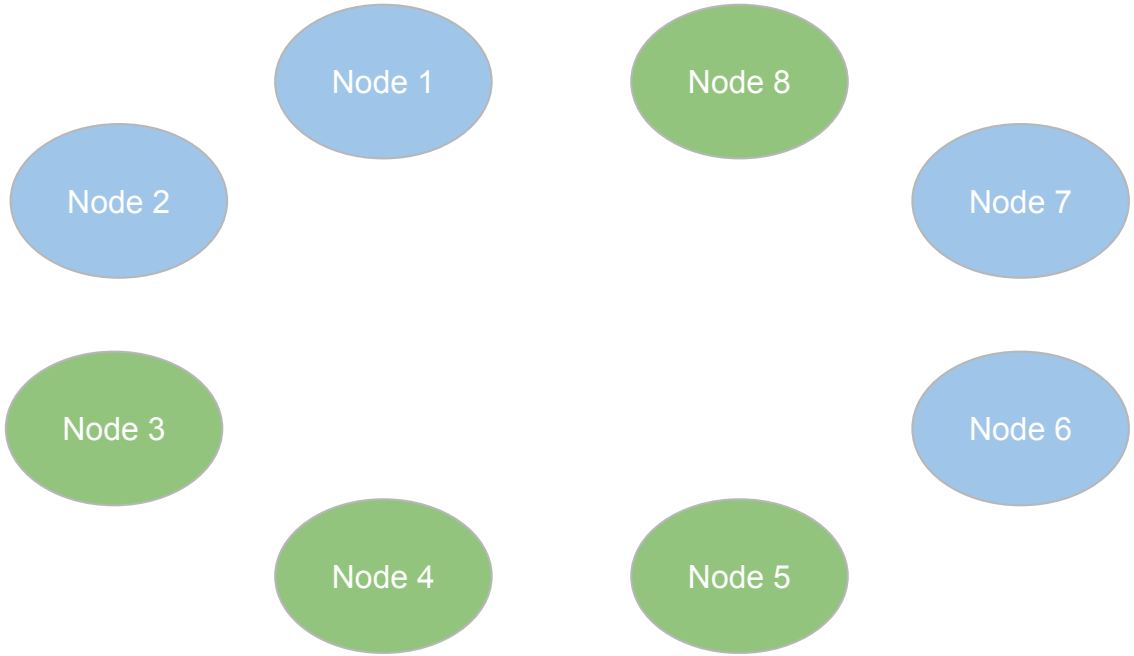
Gossip



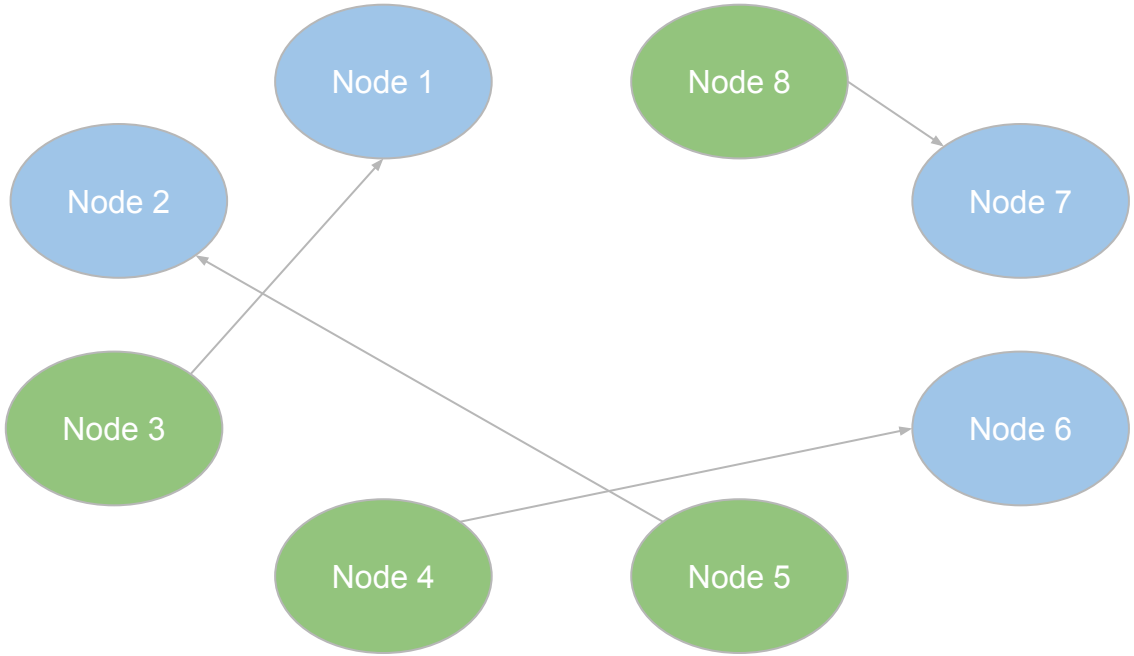
Gossip



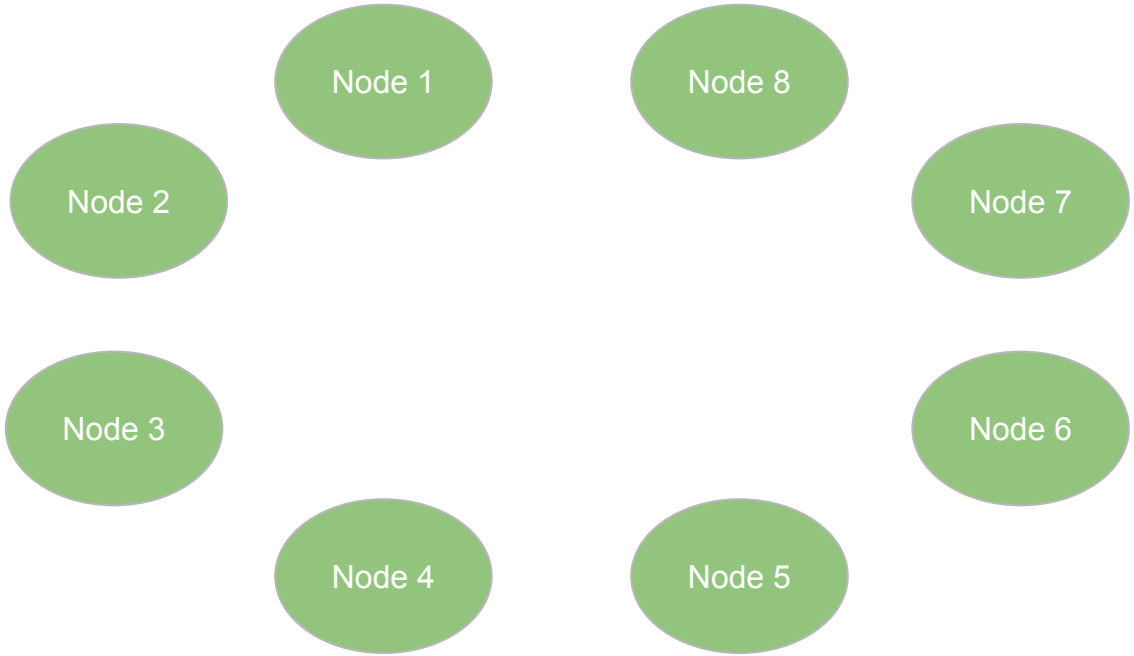
Gossip



Gossip



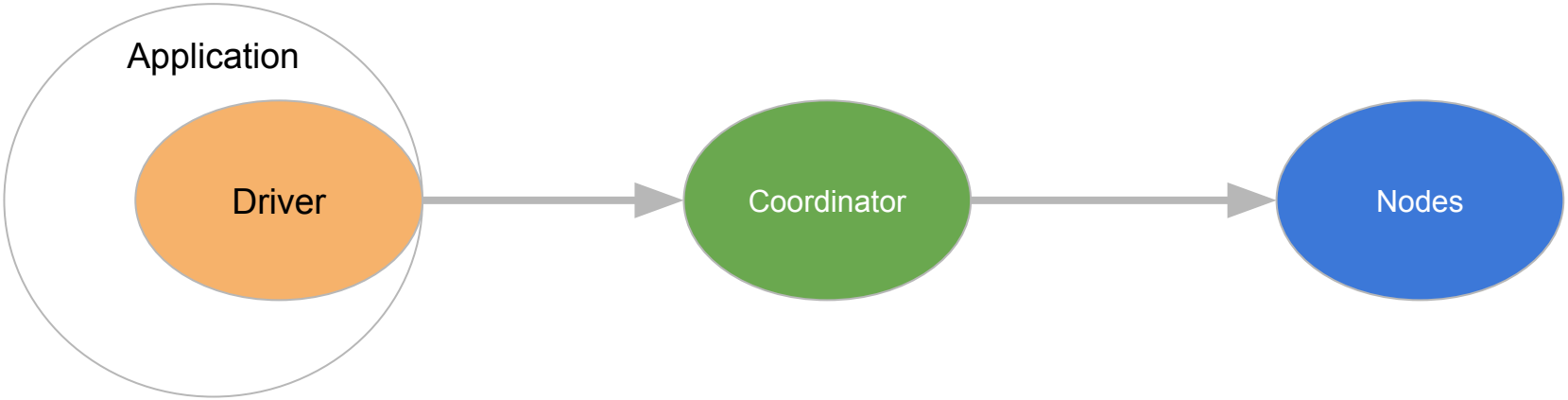
Gossip



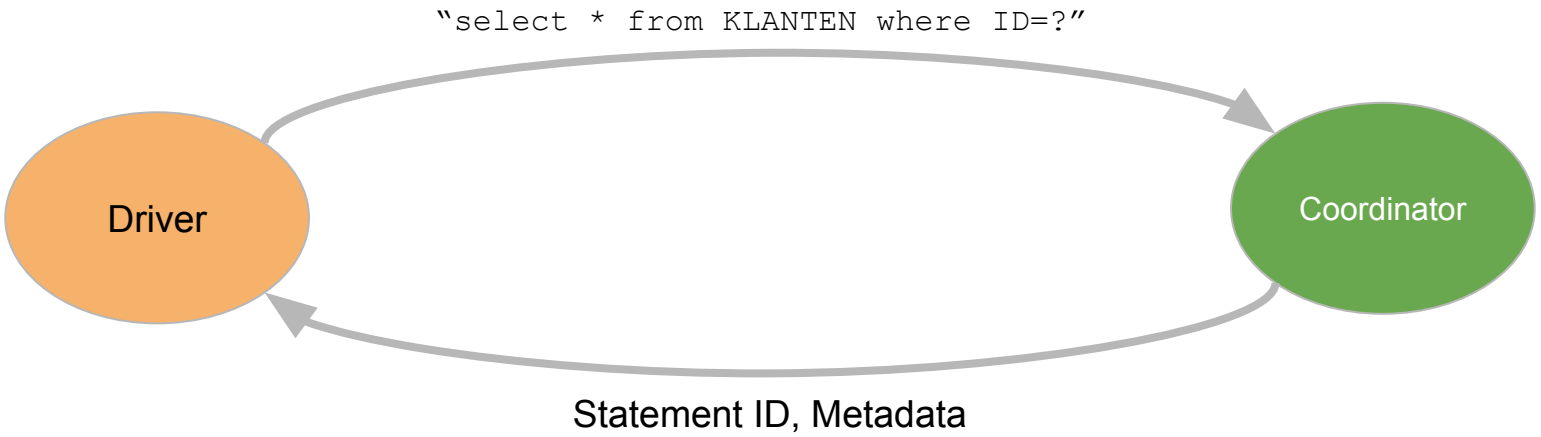
Failure detection

- heartbeat listener with the power to mark a node as down
- Gossip can only mark a node as up
- keeps backlog of timestamps intervals between updates
- periodically checks all peers

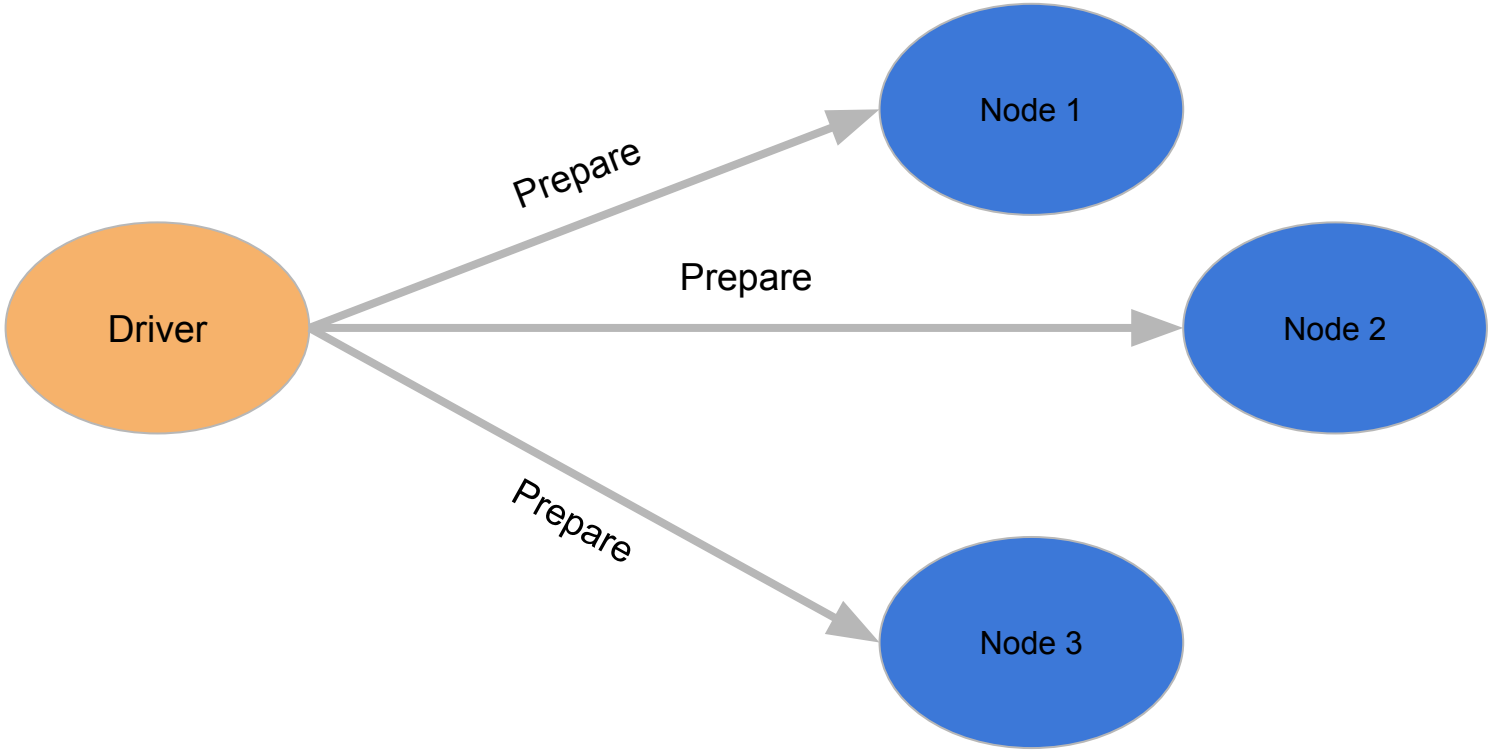
Application - Read Path



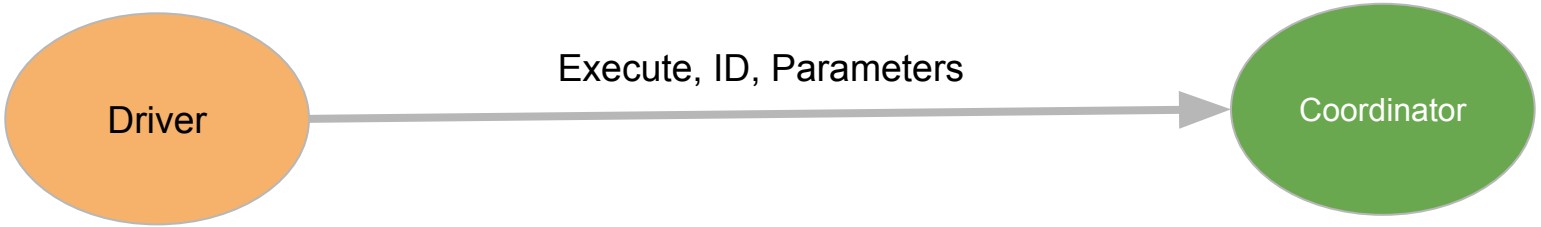
Application - Read Path step 1



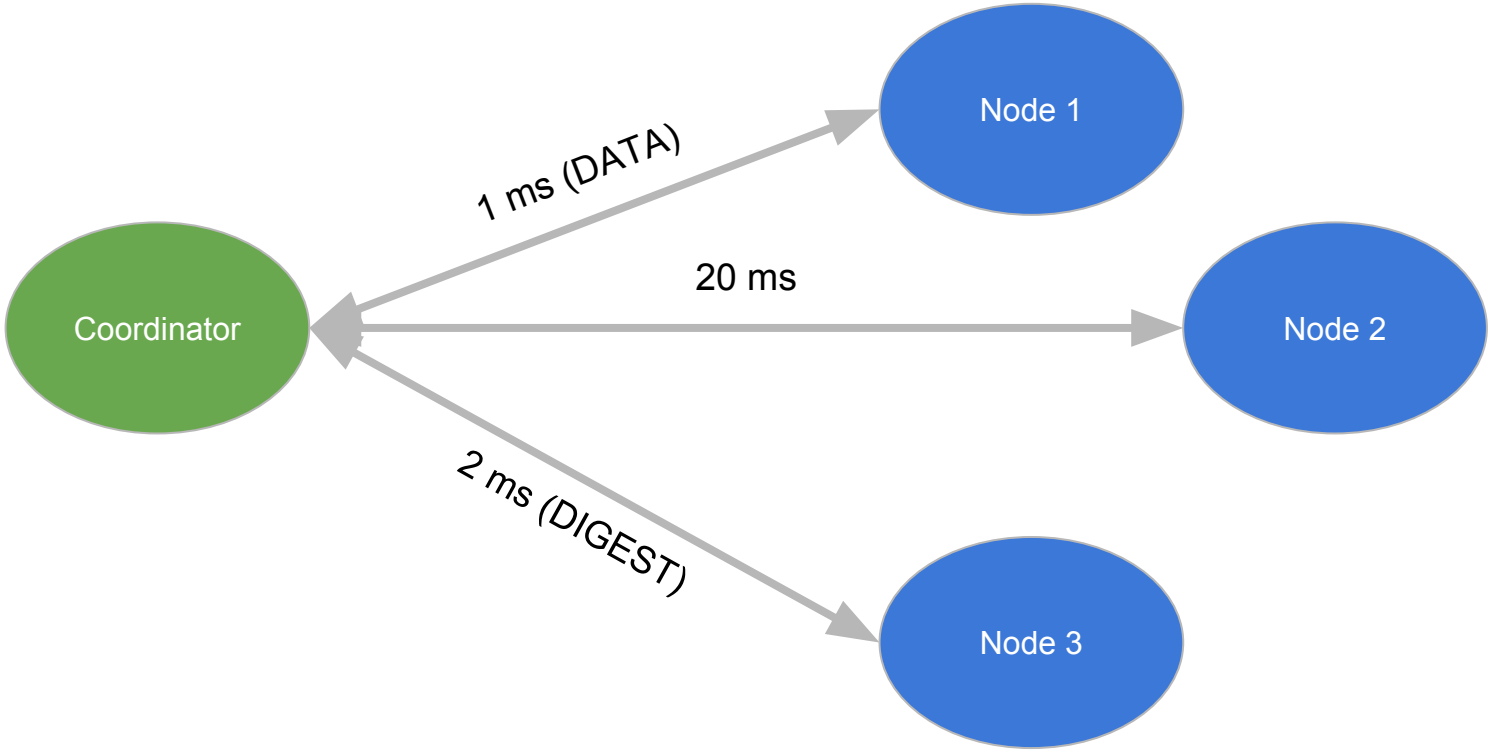
Application - Read Path step 2



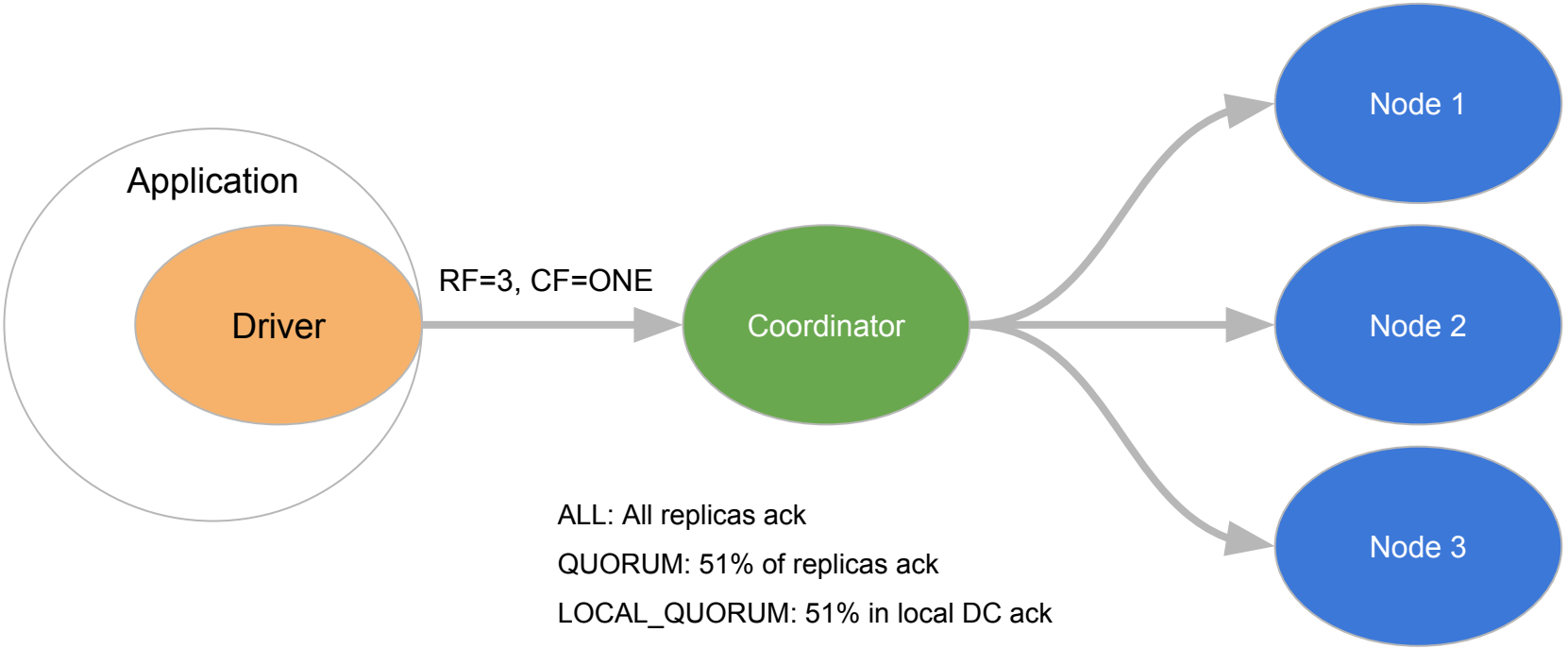
Application - Read Path step 3



Application -Read Path step 4



Consistency



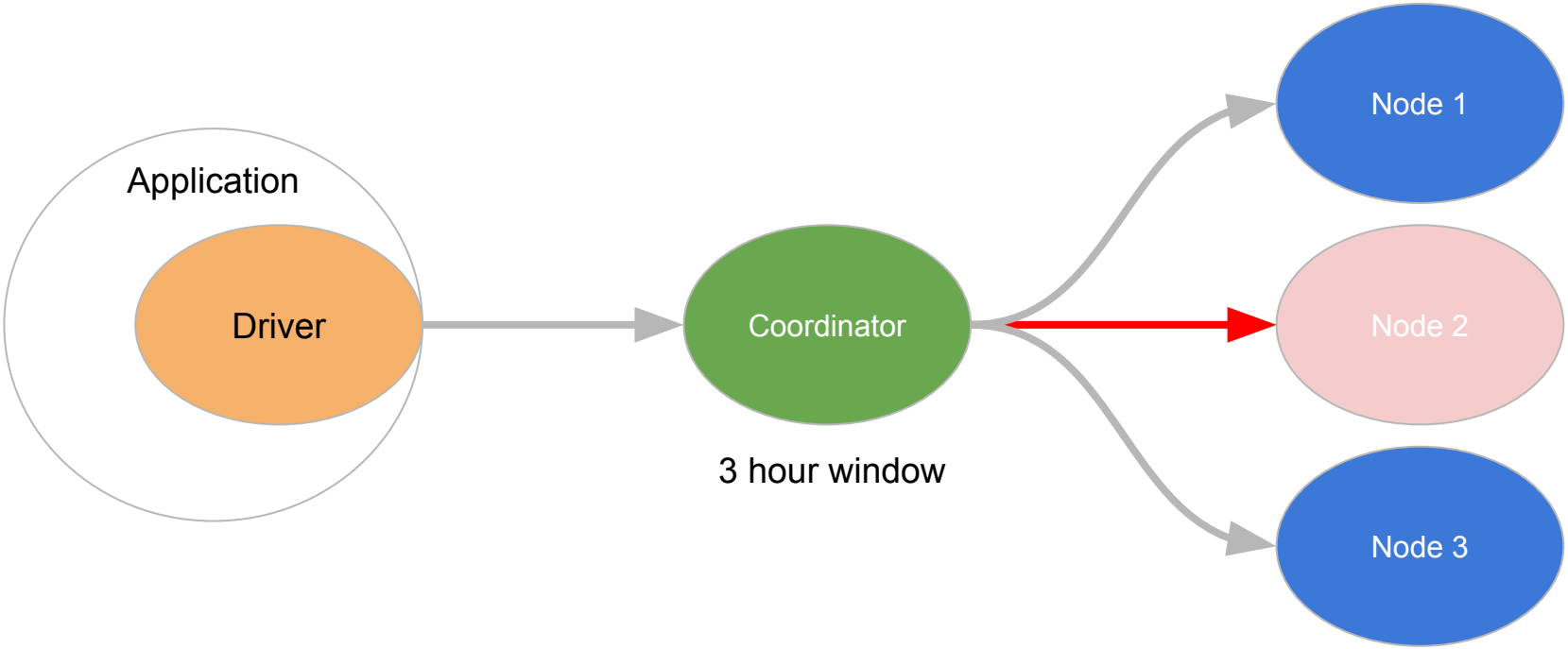
ALL: All replicas ack

QUORUM: 51% of replicas ack

LOCAL_QUORUM: 51% in local DC ack

ONE: Only one node replica acks

Hinted Handoff



Node tool

1. decommission
2. removenode
3. repair (Advised to run weekly)
4. And much more...

Sources

BigTable

- http://www.csegeek.com/csegeek/view/tutorials/algorithms/linked_list/list_intro.php
- https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/
- https://en.wikipedia.org/wiki/Binary_search_tree
- https://www.igvita.com/2012/02/06/sstable-and-log-structured-storage-leveldb/

Apache Cassandra

- Intro to Apache Cassandra
- Cassandra Internals — Understanding Gossip
- Cassandra Internals: The Read Path
- Apache Cassandra GIT repository