



NoSQL: Redis, Cass, ES, Kafka

Diego Pacheco

About Me



- ❑ Cat's Father
- ❑ Principal Software Architect
- ❑ Agile Coach
- ❑ SOA/Microservices Expert
- ❑ DevOps Practitioner
- ❑ Speaker
- ❑ Author



diegopacheco



@diego_pacheco

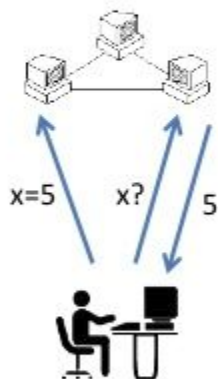


<http://diego-pacheco.blogspot.com.br/>

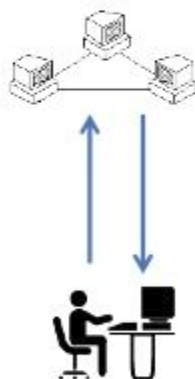


CAP Theorem

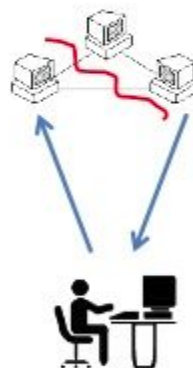
Consistency



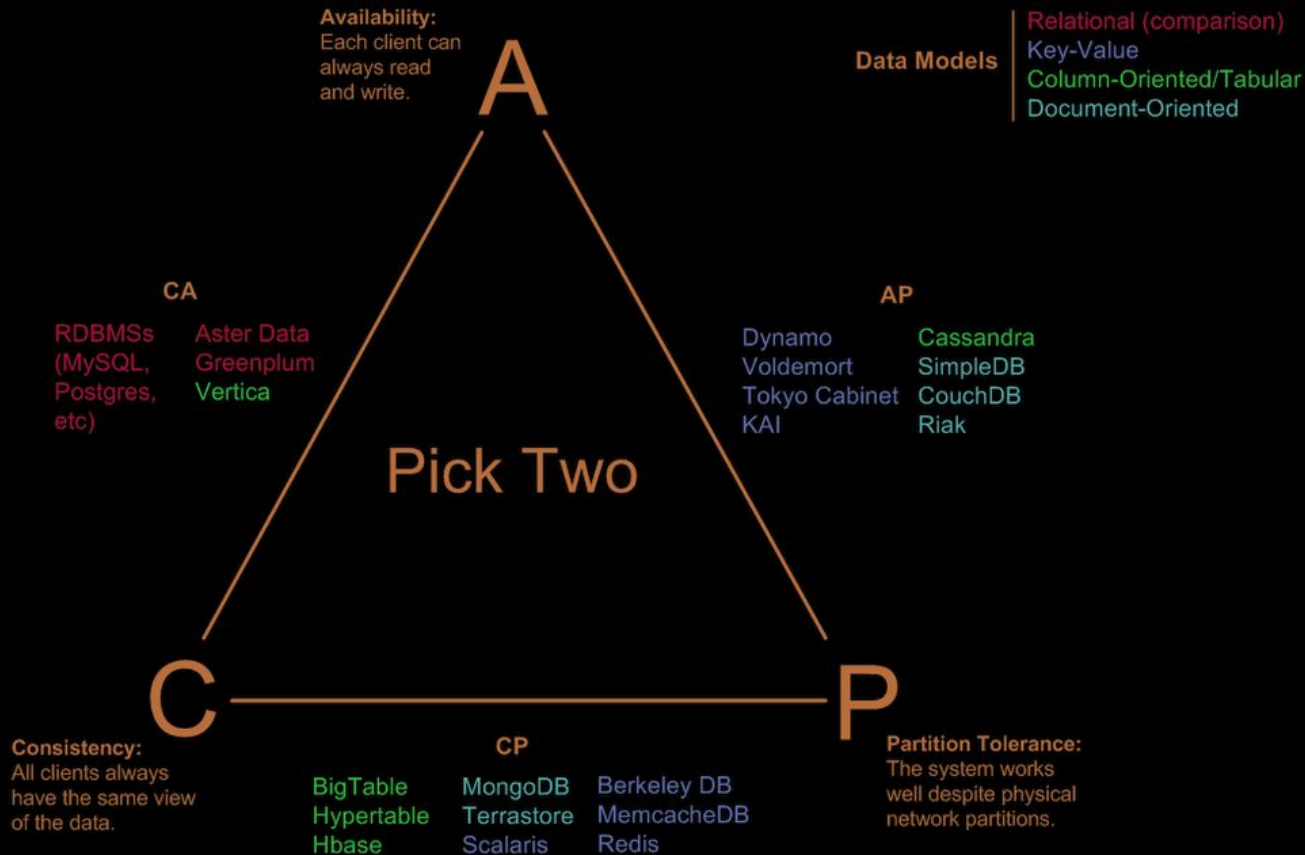
Availability



Partition tolerance



Visual Guide to NoSQL Systems



key-value

Amazon
DynamoDB (Beta)

ORACLE
BERKELEY DB 11g



graph

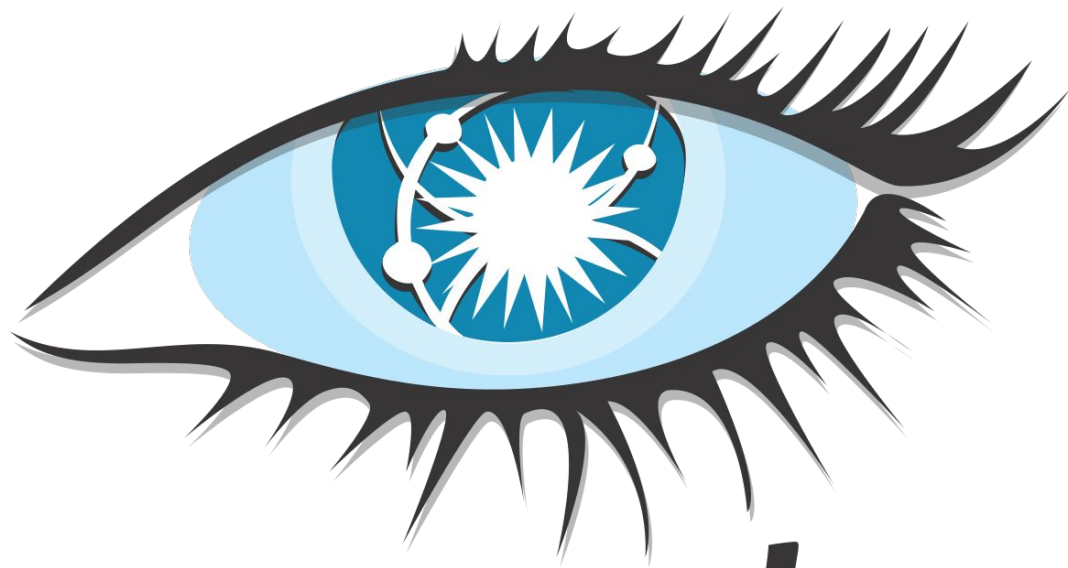


column



document



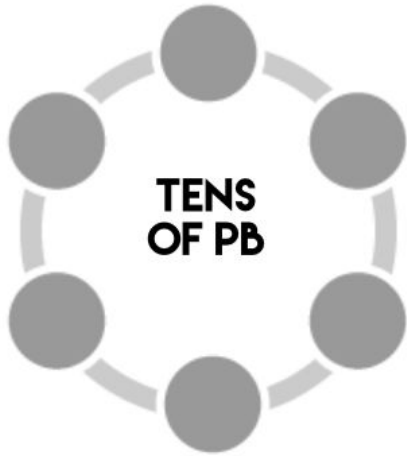


cassandra

Cassandra

- ❑ Column oriented NoSQL
- ❑ Created by Facebook for Inbox Search
- ❑ Inspired from Amazon Dynamo Paper
- ❑ Written in Java
- ❑ Symetrics Nodes – NO SPOF
- ❑ Scalable
- ❑ Awesome Write throughput
- ❑ High Availability
- ❑ Eventual Consistency
- ❑ Consistency is tunable

Cassandra Battle Tested

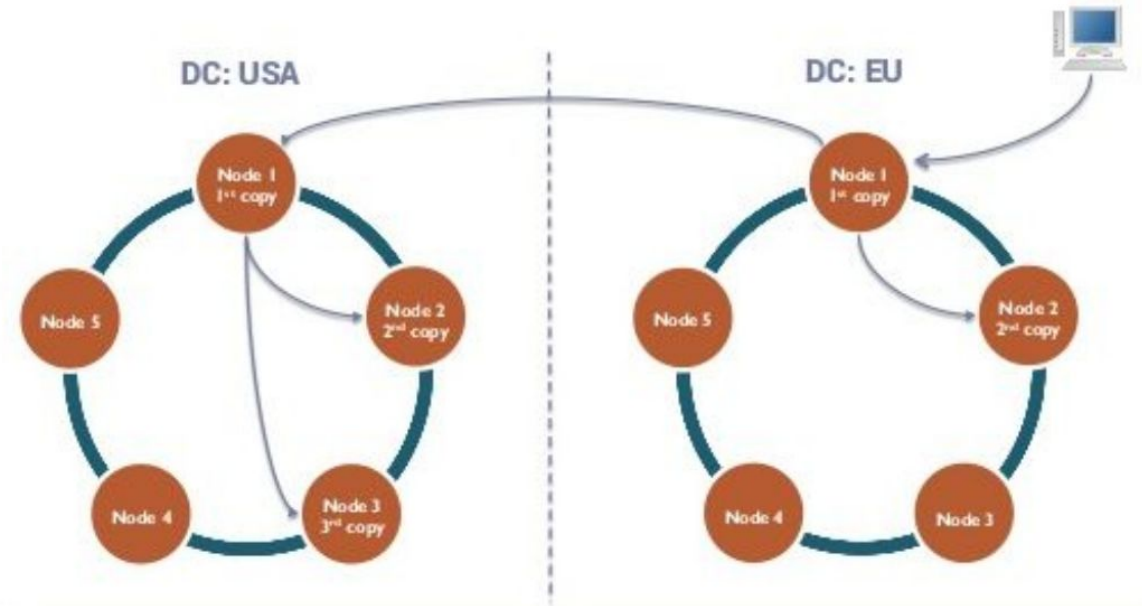


NETFLIX

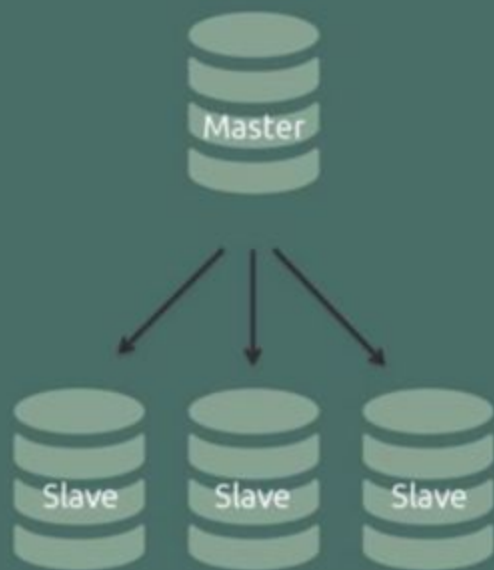


Multi-DC | Cloud Native

```
CREATE KEYSPACE johnny WITH REPLICATION =  
{'class':'NetworkTopologyStrategy', 'USA':3, 'EU': 2};
```



Partition Strategy



OR



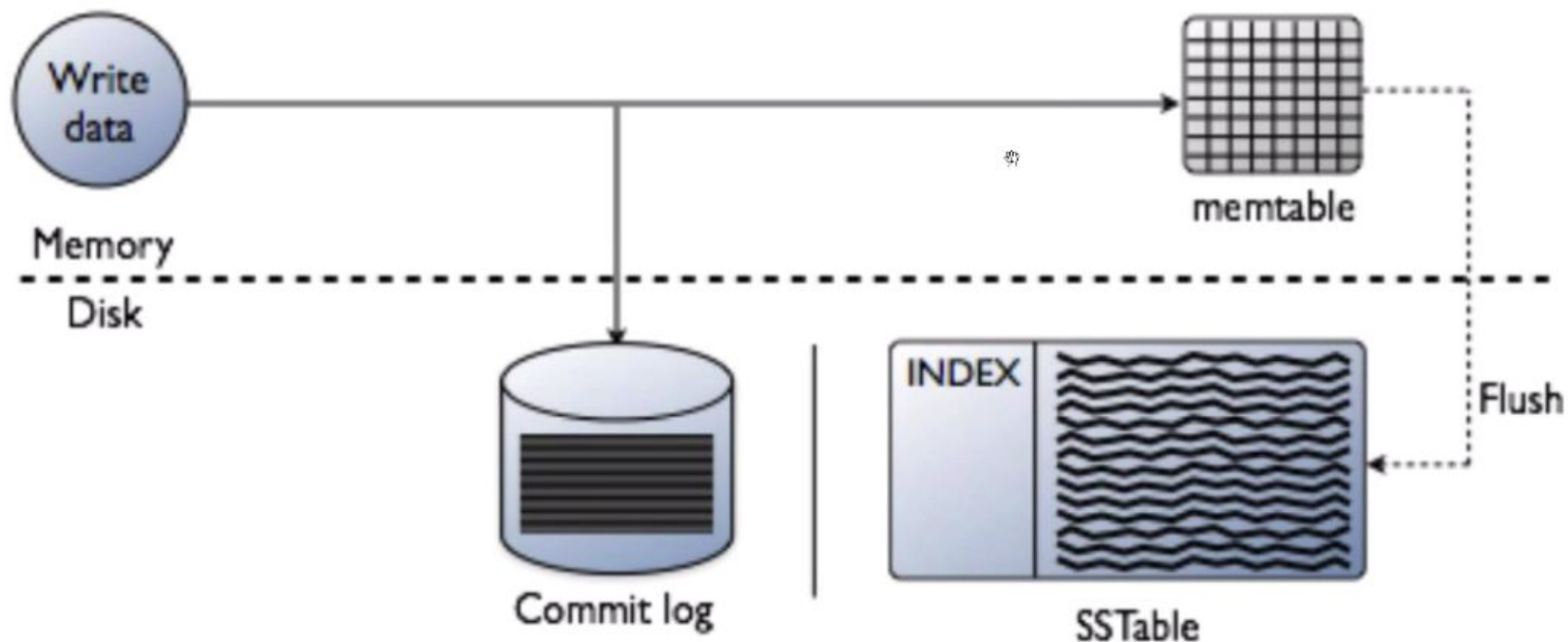
Node 1

Node 2

Node 3

Write in Cassandra

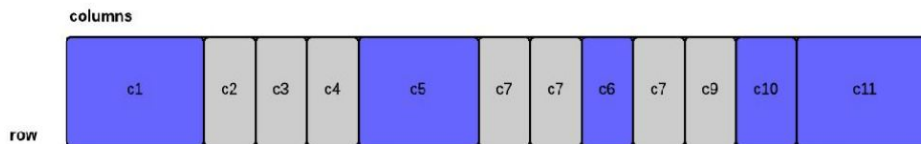
- 1 SSTable per column Family
- Bloom Filter index for Read performance



Tombstones

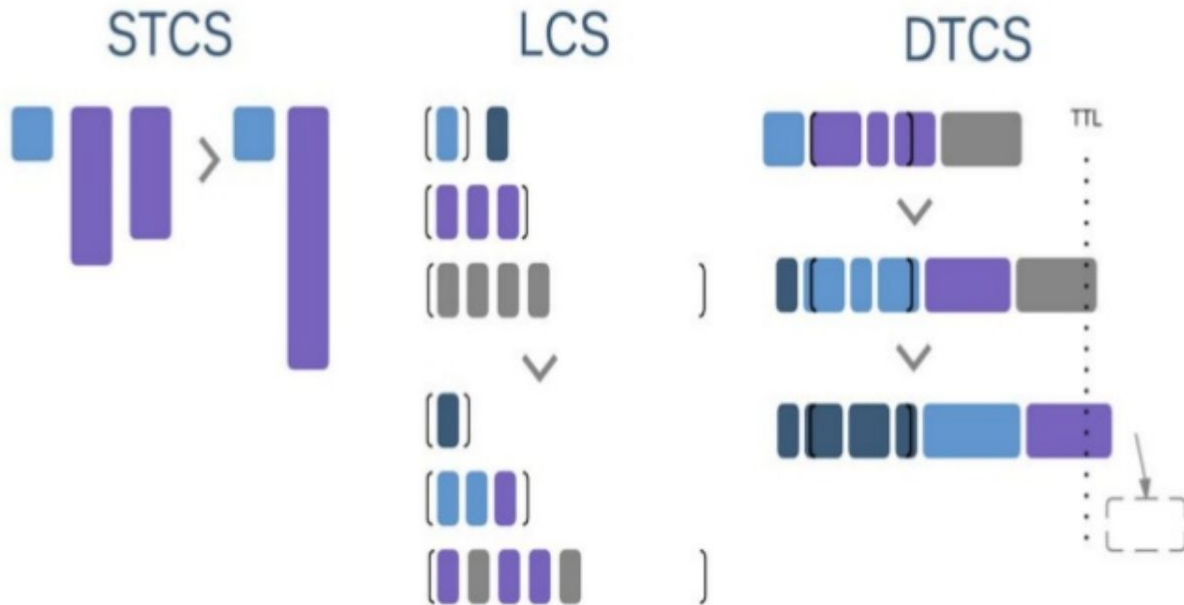


- ❑ Deleted data is MARKED as Removed == Tombstone
- ❑ Data is deleted and removed during compaction
- ❑ Compaction can happen in few days depending of the configs.
- ❑ Queries on partition with lots of tombstones requires lots of filtering which can slow down the CASS performance.
- ❑ Collections operations can lead to tombstones depending on what you do.
- ❑ There are Compaction Trade-Offs.



blue - live columns
grey - deleted columns /
tombstones

Compaction Strategy



- ❑ **STCS**
 - ❑ Default
 - ❑ Insert-Heavy
 - ❑ General Workloads
- ❑ **LCS**
 - ❑ Read Heavy
 - ❑ More Updates than Inserts
- ❑ **DTCS**
 - ❑ Time Series
 - ❑ Inserts out of order
 - ❑ Updates for old data

Anti-Patterns

Anti-Patterns

- ❑ Using Cassandra as a queue or queue-like table
 - ❑ Tombstones
 - ❑ Lots of deleted columns(expiry) and slice-queries don't play well
 - ❑ <http://www.datastax.com/dev/blog/cassandra-anti-patterns-queues-and-queue-like-datasets>
- ❑ CQL Nulls
 - ❑ Reading Tombstones
 - ❑ Write NULL create tombstones
- ❑ Intensive Updates on SAME column
 - ❑ Sensor table (ID,VALUE)
 - ❑ Physical Limits
 - ❑ Solution: Timestamp as cluster key.

Benchmark: 1 million writes per second

Per Node Activity

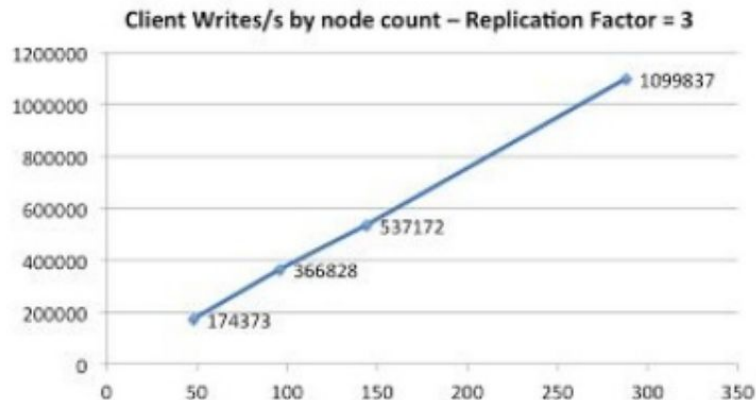
Per Node	48 Nodes	96 Nodes	144 Nodes	288 Nodes
Per Server Writes/s	10,900 w/s	11,460 w/s	11,900 w/s	11,456 w/s
Mean Server Latency	0.0117 ms	0.0134 ms	0.0148 ms	0.0139 ms
Mean CPU %Busy	74.4 %	75.4 %	72.5 %	81.5 %
Disk Read	5,600 KB/s	4,590 KB/s	4,060 KB/s	4,280 KB/s
Disk Write	12,800 KB/s	11,590 KB/s	10,380 KB/s	10,080 KB/s
Network Read	22,460 KB/s	23,610 KB/s	21,390 KB/s	23,640 KB/s
Network Write	18,600 KB/s	19,600 KB/s	17,810 KB/s	19,770 KB/s

Node specification – Xen Virtual Images, AWS US East, three zones

- Cassandra 0.8.6, CentOS, SunJDK6
- AWS EC2 m1 Extra Large – Standard price \$ 0.68/Hour
- 15 GB RAM, 4 Cores, 1Gbit network
- 4 internal disks (total 1.6TB, striped together, md, XFS)

NETFLIX

Scale-Up Linearity



NETFLIX

<http://techblog.netflix.com/2011/11/benchmarking-cassandra-scalability-on.html>

Running Cass (local)

```
diego@4winds: ~/bin/apache-cassandra-3.11.3
File Edit View Search Terminal Tabs Help

diego@4winds: ~/bin/apache-cassandra... x  diego@4winds: ~/github/diegopacheco... x  diego@4winds: ~/github/diegopacheco... x  diego@4winds: ~/bin/apache-cassandra... x  [User Icon] v

diego@4winds  ~/bin/apache-cassandra-3.11.3  bin/cassandra  14:09:22  7.89G  1.04
diego@4winds  ~/bin/apache-cassandra-3.11.3  CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.deserializeLargeSubset
(Lorg/apache/cassandra/io/util/DataInputPlus;Lorg/apache/cassandra/db/Columns;I)Lorg/apache/cassandra/db/Columns;
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubset (Ljava/util/Collection;ILorg/apache/cassandra/db/Co
lums;ILorg/apache/cassandra/io/util/DataOutputPlus;)V
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubsetSize (Ljava/util/Collection;ILorg/apache/cassandra/d
b/Columns;I)I
CompilerOracle: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSegmentManager.advanceAllocatingFrom (Lorg/apache/cassandra/db/c
ommitlog/CommitLogSegment;)V
CompilerOracle: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGetMoreContents ()Z
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stop ()V
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stopInPartition ()V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.doFlush (I)V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.writeExcessSlow ()V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.writeSlow (JI)V
CompilerOracle: dontinline org/apache/cassandra/io/util/RebufferingInputStream.readPrimitiveSlowly (I)J
CompilerOracle: inline org/apache/cassandra/db/rows/UnfilteredSerializer.serializeRowBody (Lorg/apache/cassandra/db/rows/Row;ILorg/apache/cass
andra/db/SerializationHeader;Lorg/apache/cassandra/io/util/DataOutputPlus;)V
CompilerOracle: inline org/apache/cassandra/io/util/Memory.checkBounds (JJ)V
CompilerOracle: inline org/apache/cassandra/io/util/SafeMemory.checkBounds (JJ)V
CompilerOracle: inline org/apache/cassandra/utils/AsymmetricOrdering.selectBoundary (Lorg/apache/cassandra/utils/AsymmetricOrdering;Op;II)I
CompilerOracle: inline org/apache/cassandra/utils/AsymmetricOrdering.strictnessOfLessThan (Lorg/apache/cassandra/utils/AsymmetricOrdering;Op;)I
I
CompilerOracle: inline org/apache/cassandra/utils/BloomFilter.indexes (Lorg/apache/cassandra/utils/IFilter/FilterKey;I)J
CompilerOracle: inline org/apache/cassandra/utils/BloomFilter.setIndexes (JJII)J)V
CompilerOracle: inline org/apache/cassandra/utils/ByteBufferUtil.compare (Ljava/nio/ByteBuffer;B)I
CompilerOracle: inline org/apache/cassandra/utils/ByteBufferUtil.compare (Ljava/nio/ByteBuffer;I)
CompilerOracle: inline org/apache/cassandra/utils/ByteBufferUtil.compareUnsigned (Ljava/nio/ByteBuffer;Ljava/nio/ByteBuffer;I)
CompilerOracle: inline org/apache/cassandra/utils/FastByteOperations$UnsafeOperations.compareTo (Ljava/lang/Object;JII)I
CompilerOracle: inline org/apache/cassandra/utils/FastByteOperations$UnsafeOperations.compareTo (Ljava/lang/Object;JII)I
CompilerOracle: inline org/apache/cassandra/utils/FastByteOperations$UnsafeOperations.compareTo (Ljava/lang/Object;JII)I
CompilerOracle: inline org/apache/cassandra/utils/FastByteOperations$UnsafeOperations.compareTo (Ljava/lang/Object;JII)I
CompilerOracle: inline org/apache/cassandra/utils/vint/VIntCoding.encodeVint (JI)B
INFO [main] 2018-12-28 14:09:32,187 YamlConfigurationLoader.java:89 - Configuration location: file:/home/diego/bin/apache-cassandra-3.11.3/co
nf/cassandra.yaml
INFO [main] 2018-12-28 14:09:32,732 Config.java:495 - Node configuration:[allocate_tokens_for_keyspace=null; authenticator=AllowAllAuthentica
tor; authorizer=AllowAllAuthorizer; auto_bootstrap=true; auto_snapshot=true; back_pressure_enabled=false; back_pressure_strategy=org.apache.ca
ssandra.net.RateBasedBackPressure{high_ratio=0.9, factor=5, flow=FAST}; batch_size_fail_threshold_in_kb=50; batch_size_warn_threshold_in_kb=5;
batchlog_replay_throttle_in_kb=1024; broadcast_address=null; broadcast_rpc_address=null; buffer_pool_use_heap_if_exhausted=true; cas_contenti
on_timeout_in_ms=1000; cdc_enabled=false; cdc_free_space_check_interval_ms=250; cdc_raw_directory=null; cdc_total_space_in_mb=0; client_encrypt
ion_options=<REDACTED>; cluster_name=Test Cluster; column_index_cache_size_in_kb=2; column_index_size_in_kb=64; commit_failure_policy=stop; c
```

Create Keyspace

≡ keyspace.cql x

```
1 CREATE KEYSPACE CLUSTER_TEST WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
2 USE CLUSTER_TEST;
3 CREATE TABLE TEST ( key text PRIMARY KEY, value text);
4 INSERT INTO TEST (key,value) VALUES ('1', 'works');
5 SELECT * from CLUSTER_TEST.TEST;
```

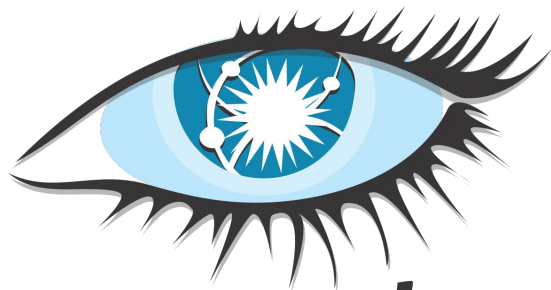
```
29 bin/nodetool status
30
31 Datacenter: datacenter1
32 =====
33 Status=Up/Down
34 |/ State=Normal/Leaving/Joining/Moving
35 -- Address      Load       Tokens     Owns (effective)  Host ID                               Rack
36 UN  127.0.0.1    204.48 KiB  256        100.0%            a9cbbaab-aa9a-4396-8d0a-1c78b8b38b17 rack1
37
38 bin/cqlsh
39
40 Connected to Test Cluster at 127.0.0.1:9042.
41 [cqlsh 5.0.1 | Cassandra 3.11.3 | CQL spec 3.4.4 | Native protocol v4]
42 Use HELP for help.
43 cqlsh> SELECT * from CLUSTER_TEST.TEST;
44
45 | key | value
46 -----+-----
47 | 1 | works
48
49 (1 rows)
50 cqlsh>
```


Go Cassandra

go-cass-sample.go x

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6
7     "github.com/gocql/gocql"
8 )
9
10 func main() {
11     cluster := gocql.NewCluster("localhost")
12     cluster.Keyspace = "cluster_test"
13     cluster.Consistency = gocql.LocalOne
14     session, _ := cluster.CreateSession()
15     defer session.Close()
16
17     var id string
18     var text string
19
20     iter := session.Query("SELECT * from cluster_test.TEST").Iter()
21     for iter.Scan(&id, &text) {
22         fmt.Println("Data:", id, text)
23     }
24     if err := iter.Close(); err != nil {
25         log.Fatal(err)
26     }
27 }
```

<https://github.com/gocql/gocql>

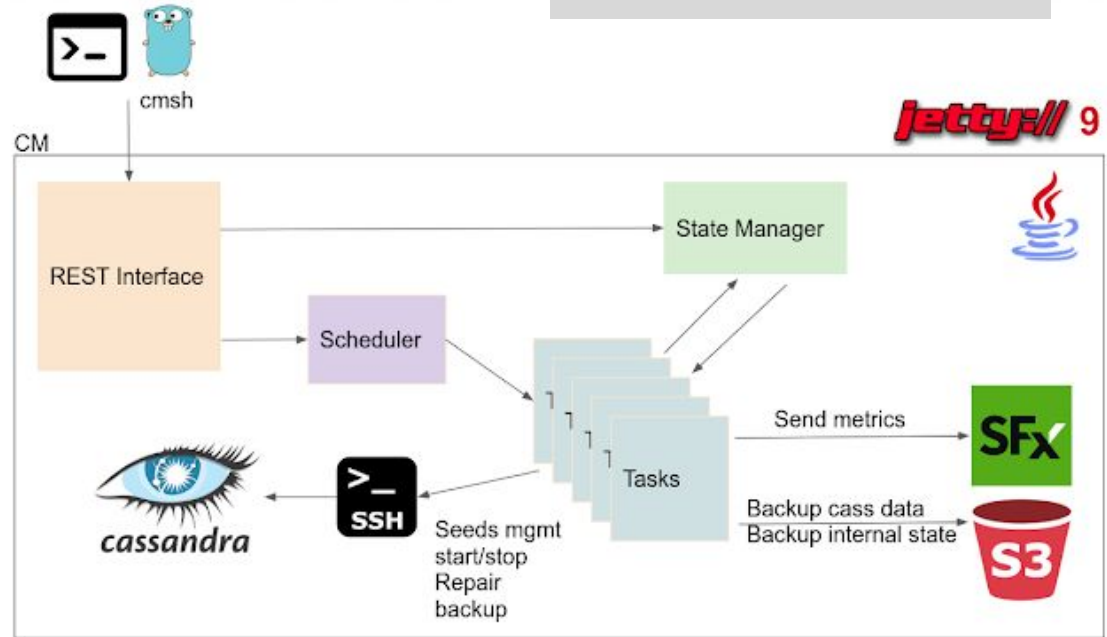
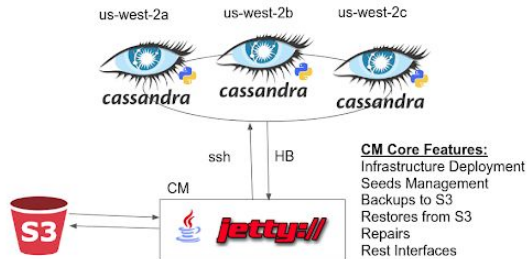


cassandra

Personal Case Study CM

CM

CM - Architecture

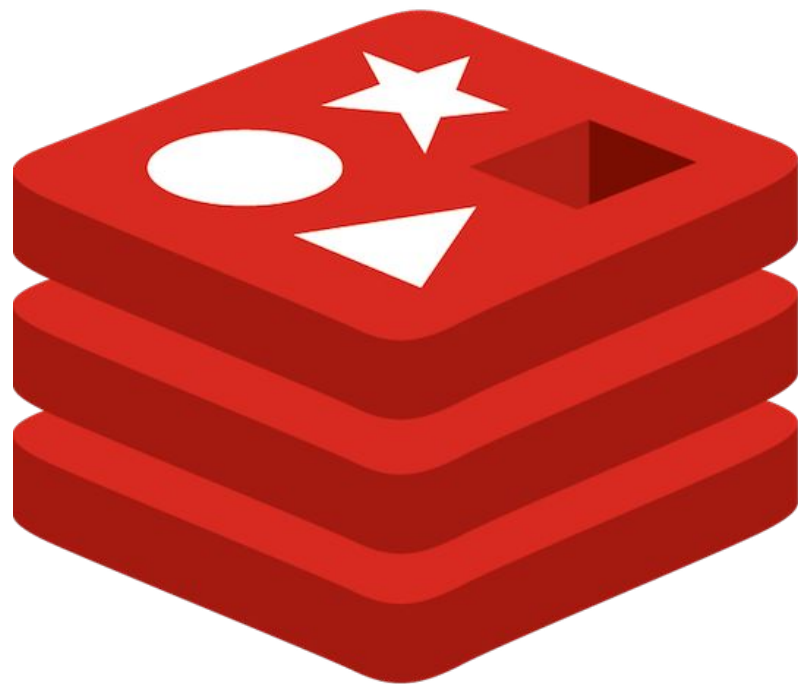


Learn & Play Local - diegopacheco/cass-docker

The screenshot shows the GitHub repository page for `diegopacheco/cassandra-docker`. The repository is described as "Cassandra cluster with Docker". It has 71 commits, 1 branch, 6 releases, 1 contributor, and is licensed under Unlicense. The repository is on the `master` branch. The file list includes:

File	Description	Commit Date
<code>.gitignore</code>	default version support	a year ago
<code>Dockerfile</code>	.	11 months ago
<code>LICENSE</code>	Initial commit	a year ago
<code>README.md</code>	.	11 months ago
<code>cassandra-2.1.19.yaml</code>	Working with cass 2.1.19 and 3.9	a year ago
<code>cassandra-3.9.yaml</code>	Working with cass 2.1.19 and 3.9	a year ago
<code>cassandra-docker-mac.sh</code>	default version support	a year ago
<code>cassandra-docker.sh</code>	improve docs	11 months ago
<code>cassandra-manager.sh</code>	.	11 months ago
<code>start-cass.sh</code>	Working with cass 2.1.19 and 3.9	a year ago

<https://github.com/diegopacheco/cassandra-docker>



Redis

- ❑ In-memory
- ❑ Key/Value store
- ❑ Written in C
- ❑ FAST
- ❑ Very robust
- ❑ Lots of specific commands / data structures
- ❑ Clients for several languages
- ❑ Redis-cluster: Master/Slave

Redis Use Cases

- ❑ Cache
- ❑ Message Queue
- ❑ Primary Data Store * -- Terms Apply
- ❑ Real Time ingestion buffer
- ❑ Handle Real Time analytics
- ❑ For management

Redis data strata

v1.0 Strings

Lists

Sets

v1.2 Sorted Sets

v2.0 Hashes

v2.2

Bit arrays

v2.8.9

HyperLogLog

v3.2

Geo Sets

Bit fields

v4

Streams (?)

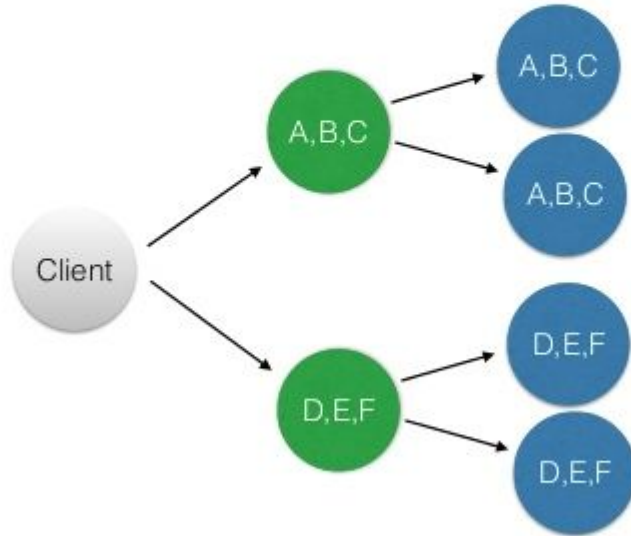
MODULES!

Redis CLI

```
1  diego@4winds:~$ redis-cli
2  127.0.0.1:6379> set users-count 0
3  OK
4  127.0.0.1:6379> set users-count 1
5  OK
6  127.0.0.1:6379> set users-count 2
7  OK
8  127.0.0.1:6379> get users-count
9  "2"
10 127.0.0.1:6379> keys *
11 1) "users-count"
12 127.0.0.1:6379>
```


Redis Cluster

Sharding and replication (asynchronous).



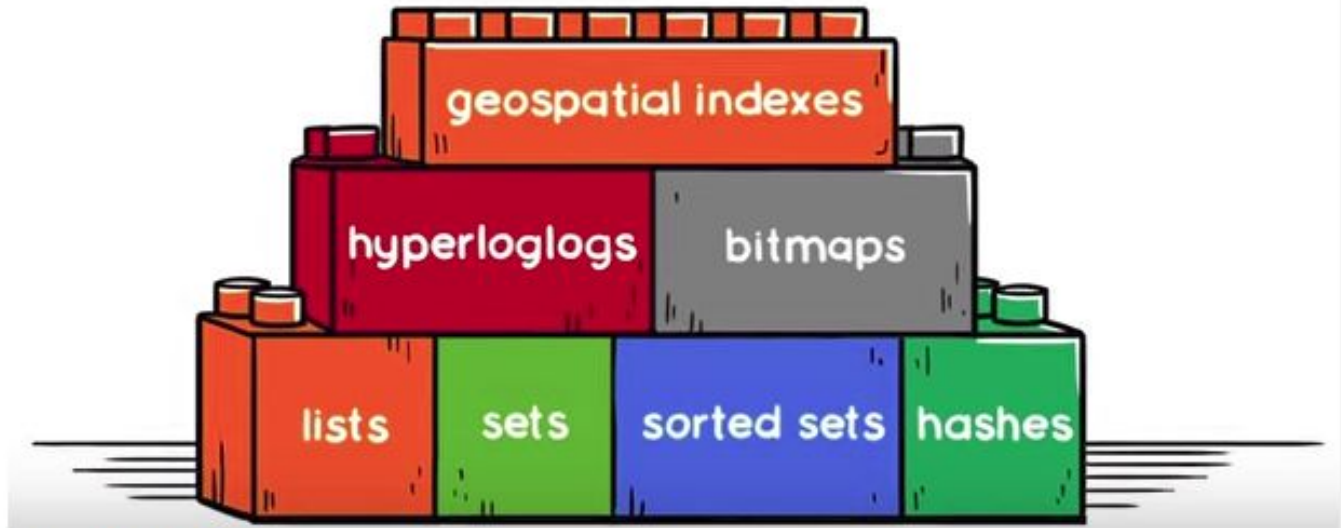
Redis Cluster Issues | Dynamite to Rescue

Redis (w/failover)

- Split-brain
- Drops all data on one component
- Not even close to consistent.



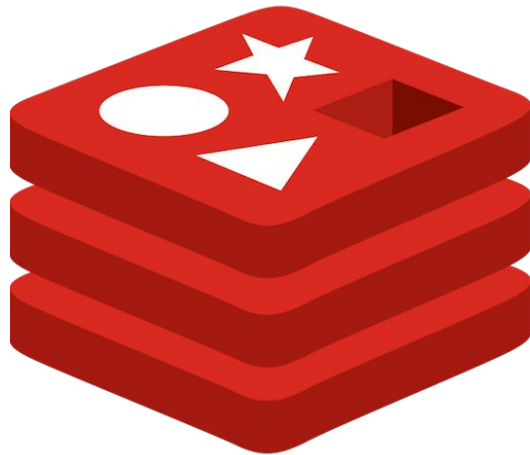
Redis data types



Picture credit : RedisLabs

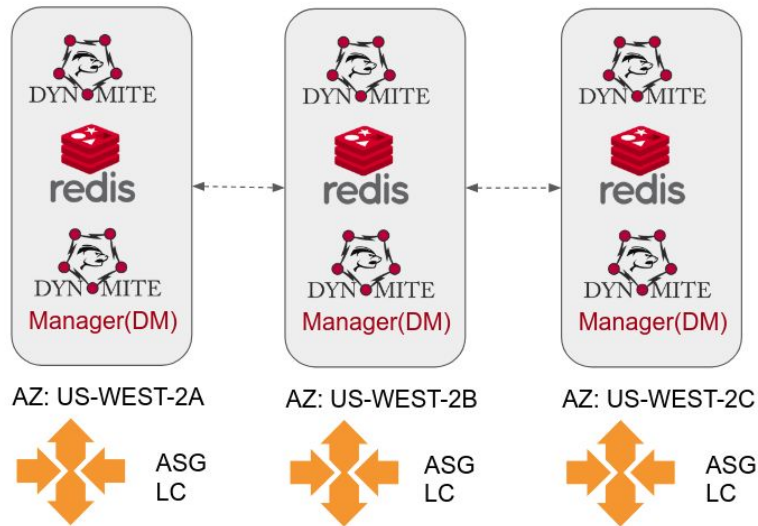
Redis Go

```
go-redis-sample.go x
1  package main
2
3  import (
4      "fmt"
5      "net/http"
6
7      "github.com/go-redis/redis"
8  )
9
10 func handle(w http.ResponseWriter, r *http.Request) {
11     client := redis.NewClient(&redis.Options{Addr: "localhost:6379"})
12     client.Incr("kcount")
13
14     val, err := client.Get("kcount").Result()
15     if err != nil {
16         panic(err)
17     }
18     result := string("key count: " + string(val))
19
20     fmt.Print(result)
21     fmt.Fprintf(w, result)
22 }
23
24 func main() {
25     fmt.Print("Serving at 0.0.0.0:9090... ")
26     http.HandleFunc("/", handle)
27     http.ListenAndServe("0.0.0.0:9090", nil)
28 }
```



Personal Case Study - Dynamite / DM use Case

Dynamite Cluster: Typical Topology



NETFLIX
OSS

https://www.youtube.com/watch?v=Z4_rzsZd70o&feature=youtu.be

Learn & Play Local - diegopacheco/dynomite-docker

The screenshot shows the GitHub repository page for `diegopacheco/dynomite-docker`. The repository is described as a "Simple Docker Image for Dynomite". It has 57 commits, 1 branch, 2 releases, 1 contributor, and is Unlicensed. The repository includes a README, a LICENSE, and several Dockerfiles and scripts. The commit history shows a series of updates to the Dockerfiles and scripts, with the latest commit on Dec 6, 2017.

diegopacheco / **dynomite-docker** Unwatch 1 Star 5 Fork 3

Code Issues 1 Pull requests 0 Projects 0 Wiki Insights Settings

Simple Docker Image for Dynomite. Edit

dynomite dcc redis docker cluster devops Manage topics

57 commits 1 branch 2 releases 1 contributor Unlicense

Branch: master New pull request Create new file Upload files Find file Clone or download

diegopacheco improved docs ... Latest commit 1299d43 on Dec 6, 2017

<code>.gitignore</code>	Added vagrant files to ignore list	2 years ago
<code>Dockerfile</code>	Kept 1 redis-conf for mac/linux	a year ago
<code>LICENSE</code>	Initial commit	2 years ago
<code>README.md</code>	improved docs	a year ago
<code>Vagrantfile</code>	Added java 8	2 years ago
<code>dynomite-docker-mac.sh</code>	Added redis port mapping for all docker clusters in mac	a year ago
<code>dynomite-docker.sh</code>	Added keys_single command	a year ago
<code>redis.conf</code>	Open Redis port for MAC and change persistence	a year ago
<code>redis_cluster_1.yml</code>	Make dynomite-docker work with code improvements and also change netw...	a year ago
<code>redis_cluster_2.yml</code>	Make dynomite-docker work with code improvements and also change netw...	a year ago
<code>redis_cluster_21.yml</code>	Make dynomite-docker work with code improvements and also change netw...	a year ago

<https://github.com/diegopacheco/dynomite-docker>



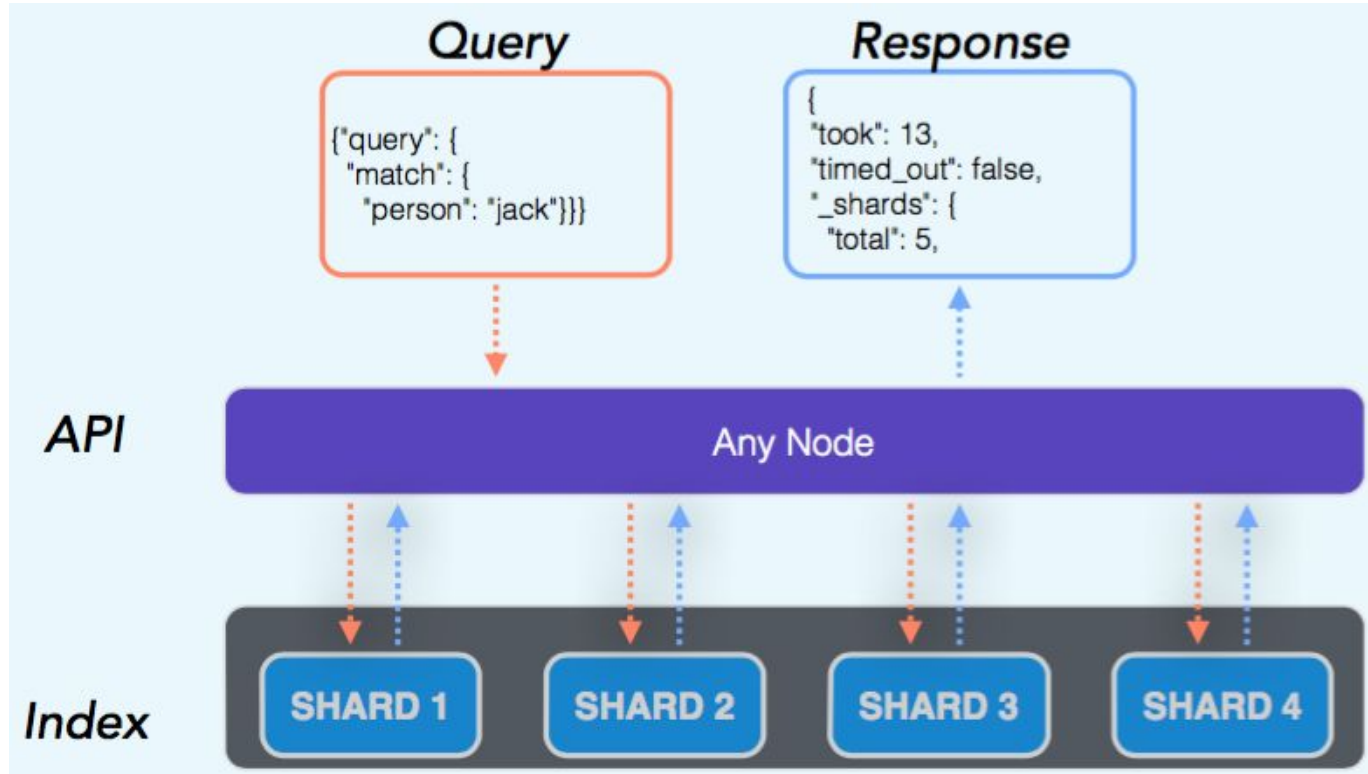
elasticsearch

ES

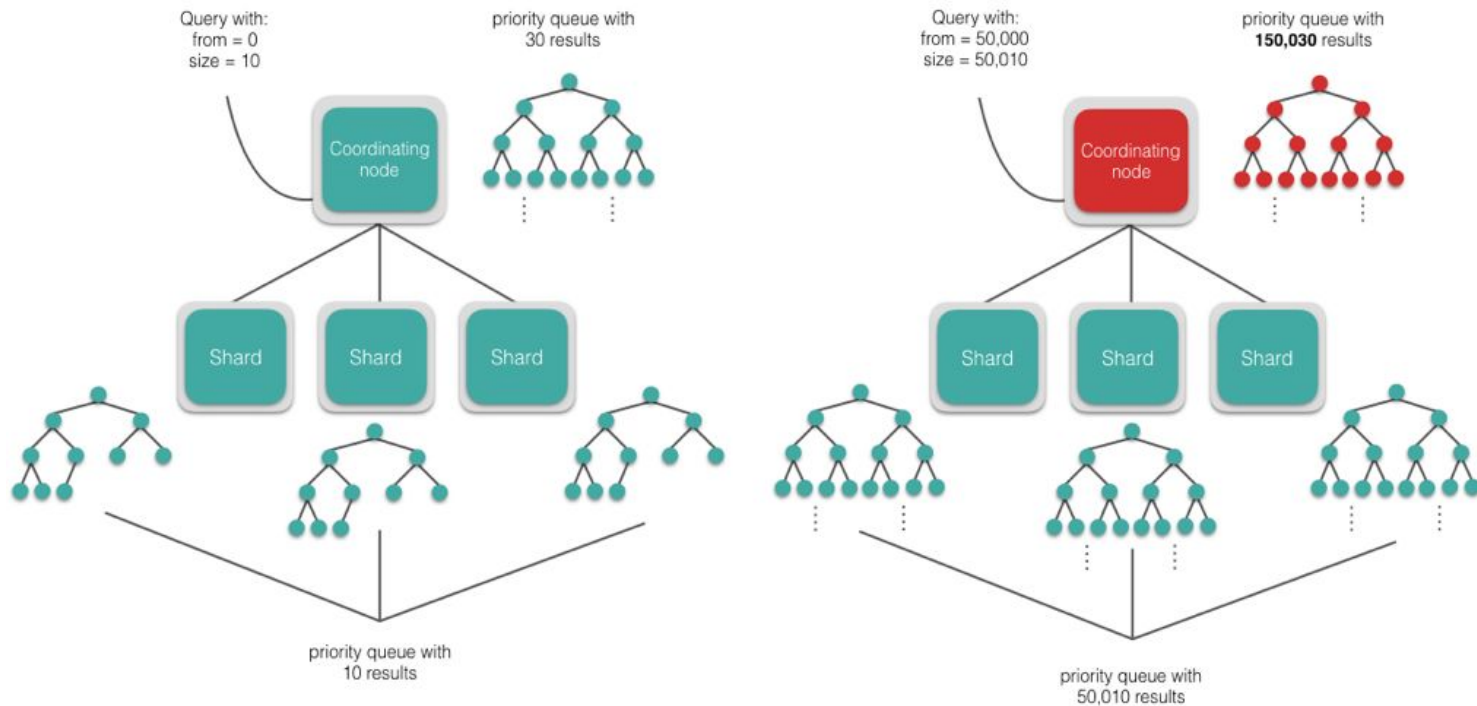


- ❑ 2010
- ❑ Cluster
- ❑ Uses Lucene Engine(Apache)
- ❑ Full Text Search Engine
- ❑ Documents (JSON) - Indexing / Feeding Process
- ❑ Distributed
- ❑ Geo Processing Support - Geo Search
- ❑ It can't be used as Source of Truth - Because loses data(Split Brain).

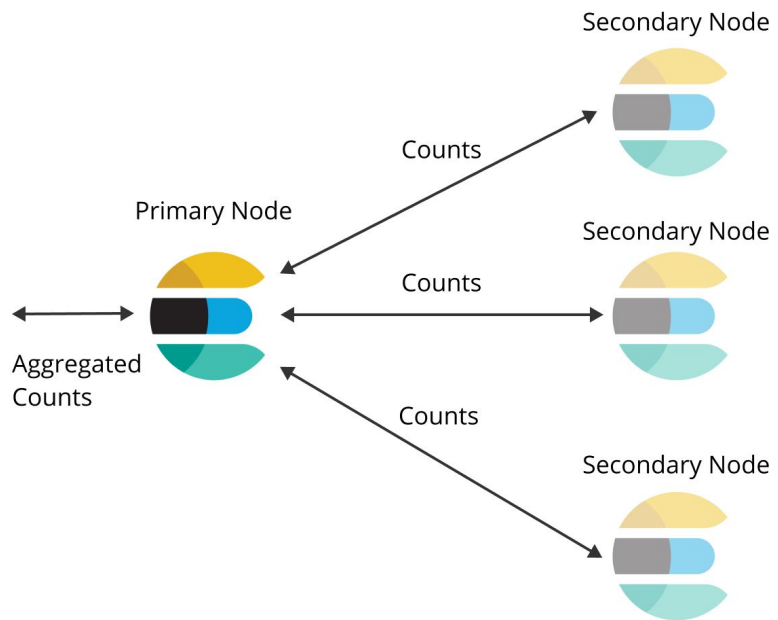
REST API



ES Cluster



ES Cluster



ES Cluster - Create Docs

```
POST localhost:9200/accounts/person/1
{
  "name" : "John",
  "lastname" : "Doe",
  "job_description" : "Systems administrator and Linux specialist"
}
```

The response will return information about the document creation:

```
{
  "_index": "accounts",
  "_type": "person",
  "_id": "1",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "created": true
}
```

ES Cluster - Retrieve it

```
GET localhost:9200/accounts/person/1
```

The result will contain metadata and also the full document (shown in the `_source` field) :

```
{
  "_index": "accounts",
  "_type": "person",
  "_id": "1",
  "_version": 1,
  "found": true,
  "_source": {
    "name": "John",
    "lastname": "Doe",
    "job_description": "Systems administrator and Linux specialist"
  }
}
```

ES Cluster - Search

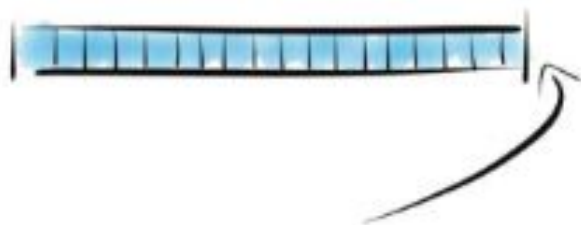
```
GET localhost:9200/_search?q=john
```

This search will return both documents, since both of them include *john*.

```
{
  "took": 58,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 2,
    "max_score": 0.2876821,
    "hits": [
      {
        "_index": "accounts",
        "_type": "person",
        "_id": "2",
        "_score": 0.2876821,
        "_source": {
          "name": "John",
          "lastname": "Smith",
          "job_description": "Systems administrator"
        }
      },
      {
        "_index": "accounts",
        "_type": "person",
        "_id": "1",
        "_score": 0.28582606,
        "_source": {
          "name": "John",
          "lastname": "Doe",
          "job_description": "Systems administrator and Linux specialist"
        }
      }
    ]
  }
}
```



KAFKA's Distributed Log



Append Only



Linear Scans

Kafka Partitions

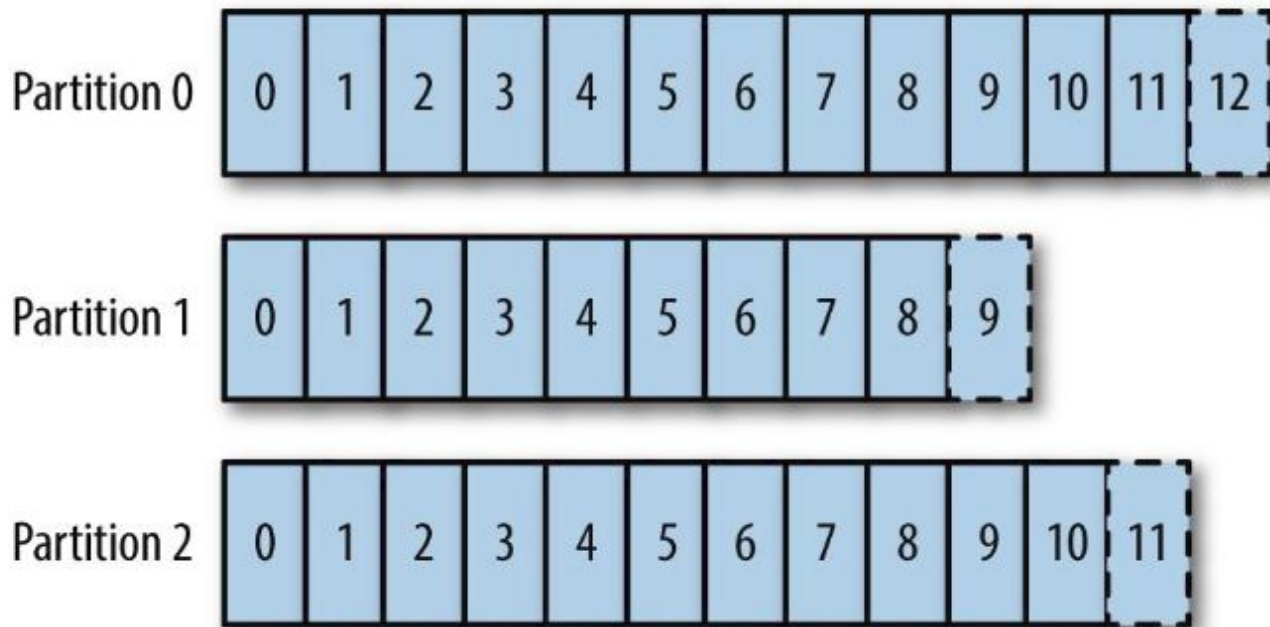


Figure 2-9. By partitioning the log, we allow each partition to act independently of all other partitions. This lets us horizontally scale the write throughput.

Log Compaction

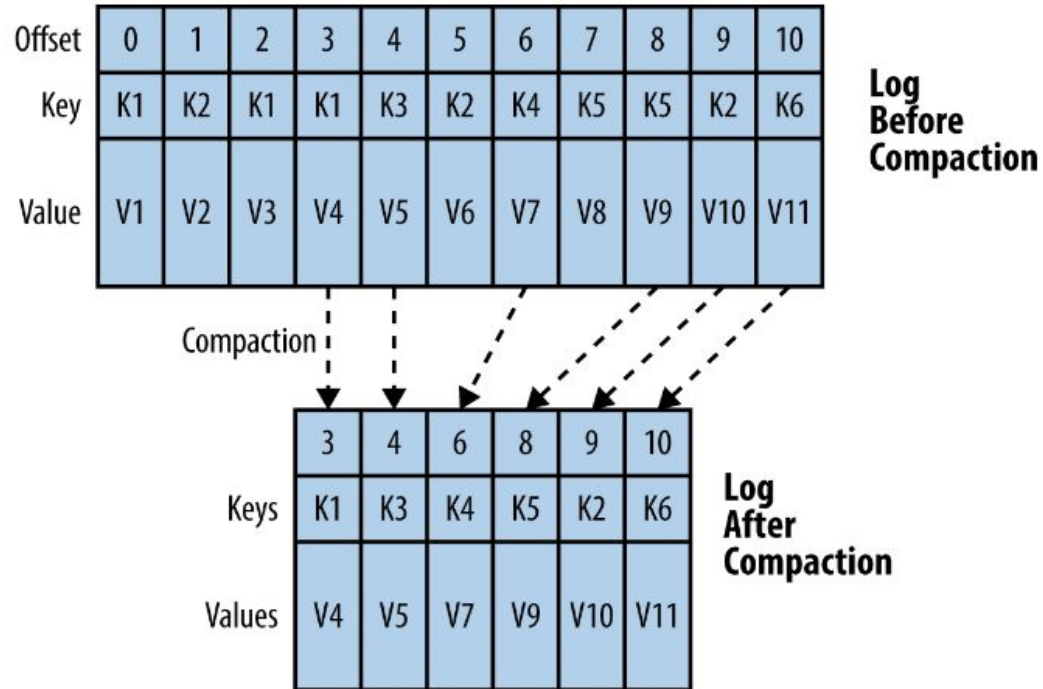
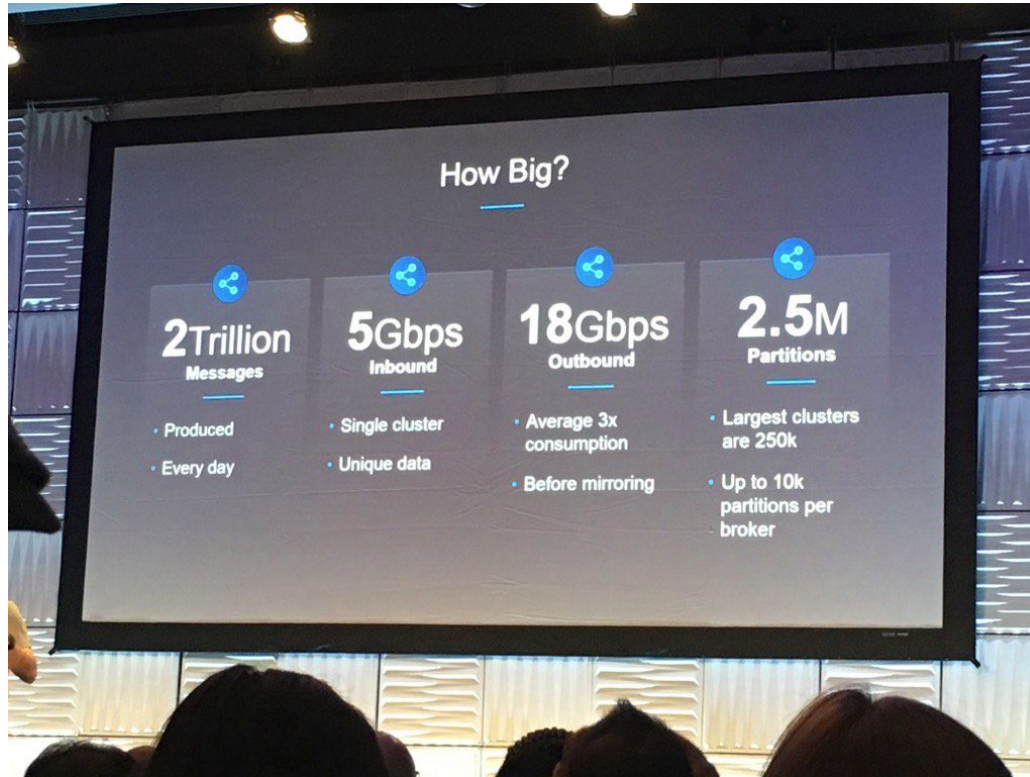
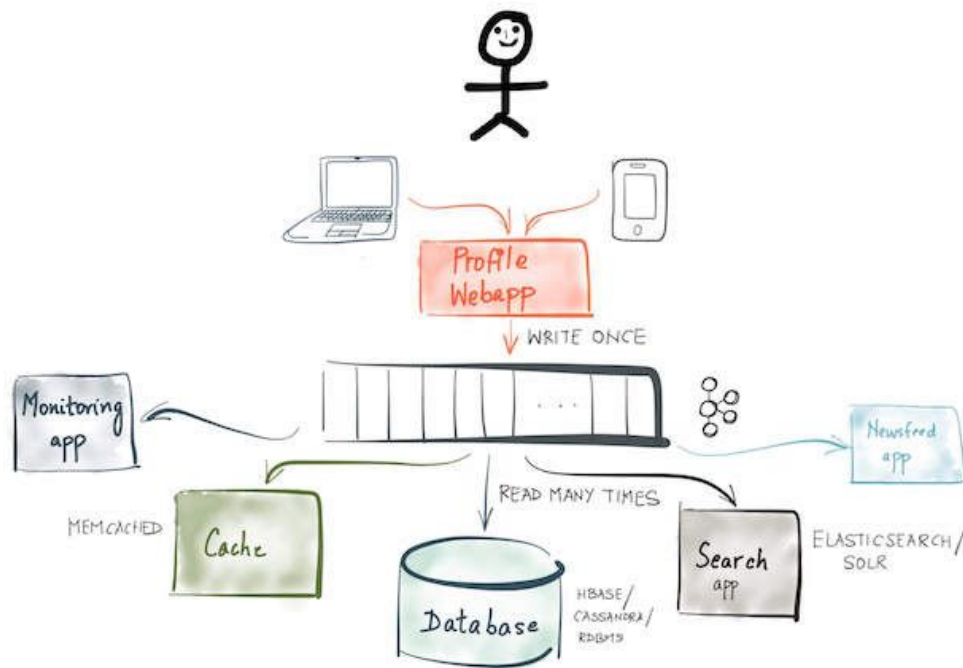


Figure 3-5. Log compaction ensures that the log retains only the latest update for each key. This is useful for modeling updates to mutable data as a log.

Kafka LinkedIn Numbers 2017



CQRS / ES



1) How we do aggregate queries without a central DB?

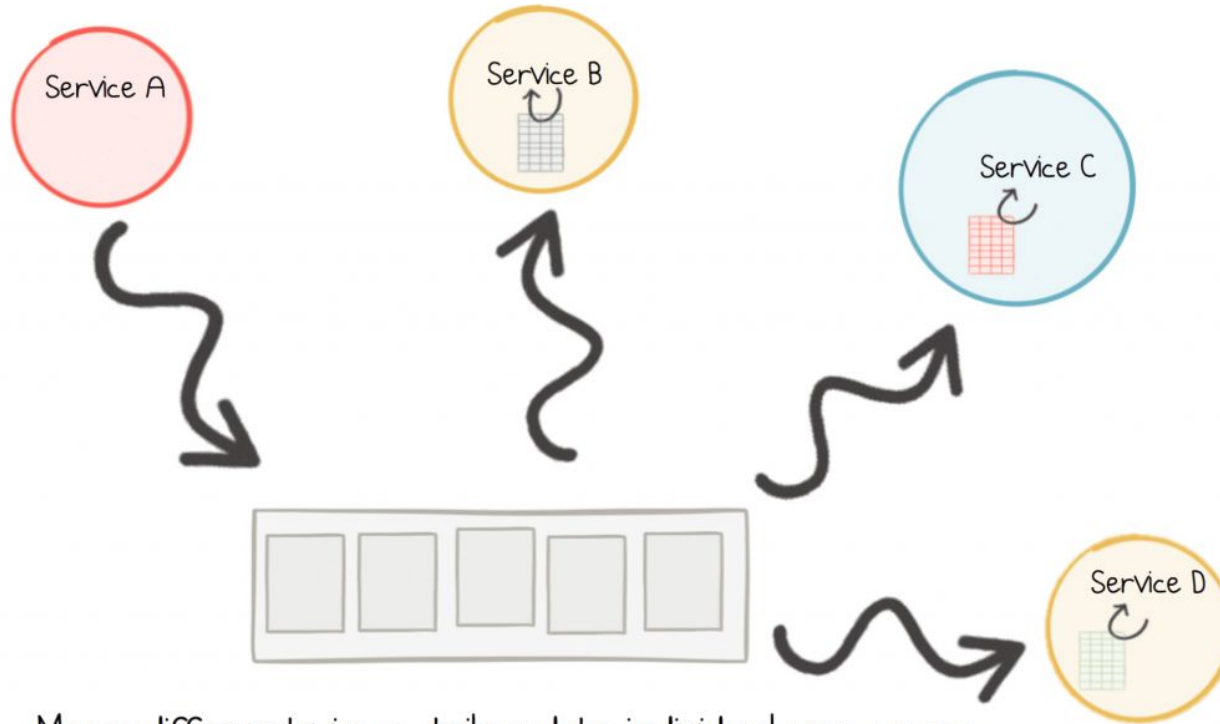
2) How do we propagate and notify changes?



Kafka and Microservices



Google Cloud Pub/Sub



Many different views, tailored to individual use cases

Learn & Play Local

Having fun with Kafka

Installing Kafka

```
sudo wget http://www.eu.apache.org/dist//kafka/0.8.2.0/kafka_2.10-0.8.2.0.tgz
tar -xzf kafka_2.10-0.8.2.0.tgz
cd kafka_2.10-0.8.2.0
```

START Zookeeper and KAFKA SERVER

```
sudo nohup bin/zookeeper-server-start.sh config/zookeeper.properties &
sudo nohup bin/kafka-server-start.sh config/server.properties &
```

Create a Topic and List all Topics

```
sudo bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
sudo bin/kafka-topics.sh --list --zookeeper localhost:2181 &
```

Create and send message to kafka topic

```
sudo bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
```

Start a consumer and consume all messages from beginning

```
sudo bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test --from-beginning
```



Apache
Zookeeper

<https://gist.github.com/diegopacheco/b1979c18357e57cc8218>



NoSQL: Redis, Cass, ES, Kafka

Diego Pacheco