MongoDB

Nathan Tippy
OCI

CAP Theorem (pick two)

Best effort categorization (not comprehensive)

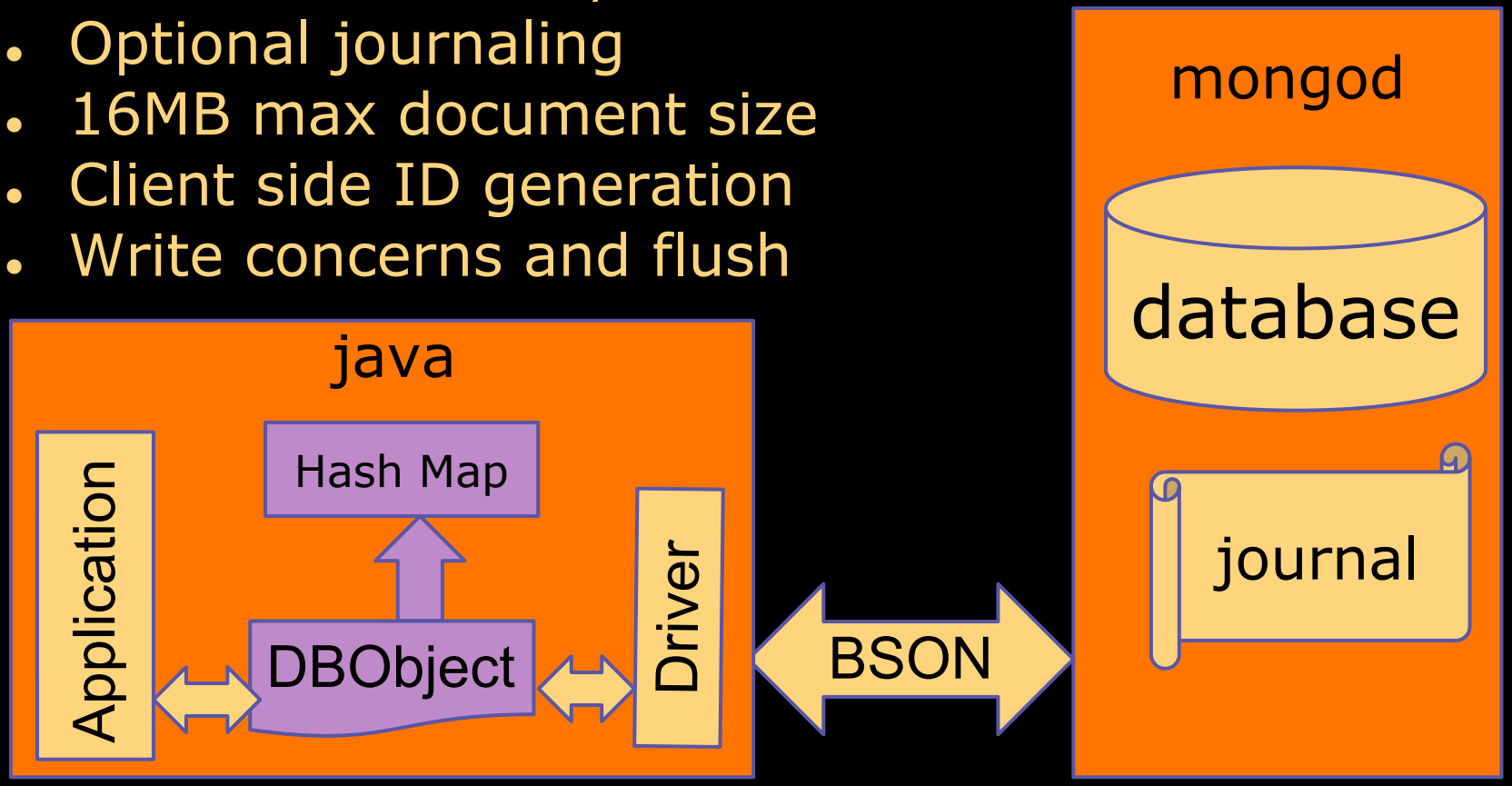
DataModel	Consistent/Available	Available/Partitioned	Partitioned /Consistent
Document		CouchDB SimpleDB Riak	MongoDB Terrastore MarkLogic
Column	Vertica	Cassandra	BigTable HyperTable HBase (Hadoop)
Key-Value		Dynamo Voldemort Tokyo Cabinet	BerkeleyDB
Graph		FlockDB Neo4j HyperGraphDB	InfoGrid InfiniteGraph
Memory			MemcacheDB Redis
Relational	MS SQL Postgres MySQL		

MongoDB Overview

- Schema-less Document Oriented (JSON)
- Consistent and Partition tolerant of CAP
- Asynchronous writes
- Replication and sharding
- Excellent performance
- Support for the traditional notion of indexes
- Atomic document updates
- No transactions and no joins
- Mongo terminal client

How does it work?

- Memory mapped DB
- Driver does BSON/JSON conversion
- Optional journaling
- 16MB max document size
- Client side ID generation
- Write concerns and flush



WriteConcern

WriteConcern can be set for the db, collection and/or each write operation.

- None - Fire and forget
- Normal - Network exceptions only
- Safe - Write to disk now
- FsyncSafe - Wait for next flush
- JournalSafe - Wait for write to journal
- ReplicasSafe - 2 members of cluster
- Majority - Majority of cluster

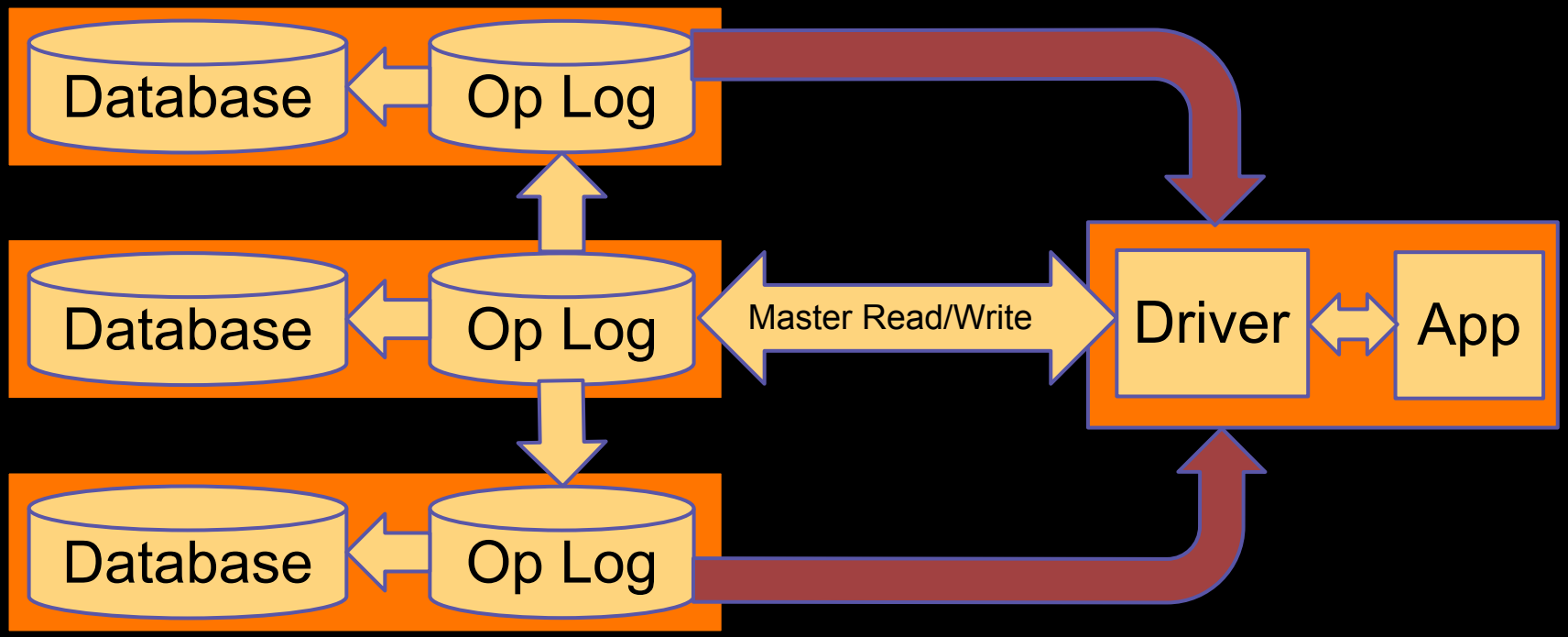
Object Id

"_id" : ObjectId("4e7fe13a83a26cdc0f203496")

- Generated by the driver under '_id'
- 12 Byte value used as the primary key
- Stored as big endian
  - Bytes 0, 1, 2, 3 (seconds since 1970)
  - Bytes 4, 5, 6 (Machine id)
  - Bytes 7, 8 (Process id)
  - Bytes 9, 10, 11 (incrementing or random)

If you decided to generate your own _id be careful to ensure each unique document has a unique immutable _id value.

Replica Sets



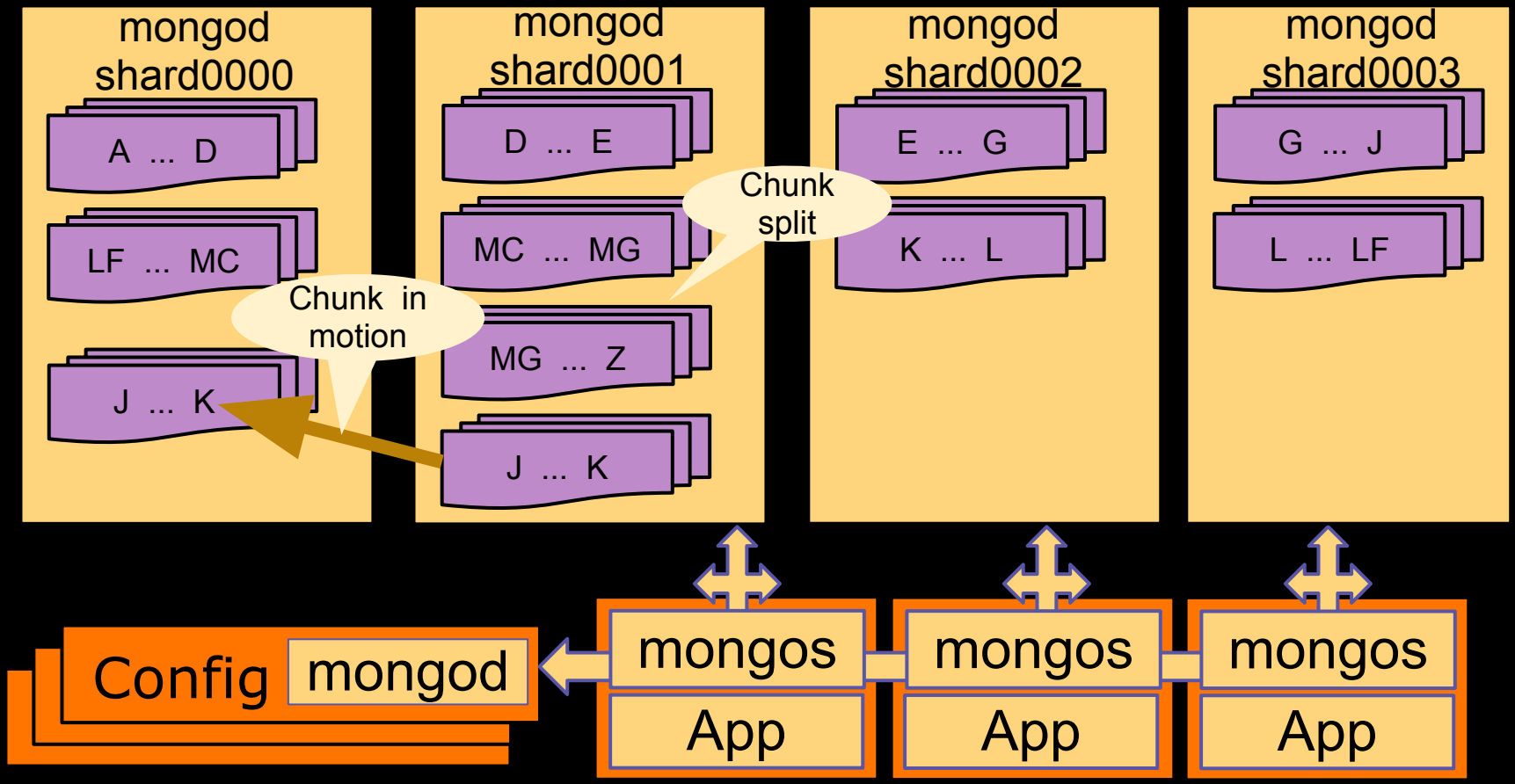
- Master elected automatically
- All writes go to master
- SlaveOK enforces read from slaves

Replica Sets

- Redundancy across remote machines
- Hot fail-over, Hot scalability
- Driver finds the master and slaves
- CP or AP your choice and your responsibility
- Each connection will round robin to the next slave
  - Large results sets are stuck on one server
- Writes can be very expensive
  - The reads may be blocked by the replication
- An odd number of members is highly recommended.

Sharded Cluster

- Distributed performance for reads, writes and map-reduce
- Scale to 2^64B or 18 Exabytes (18,446,744,073,709,551,616 B)



Sharded Cluster

- Each shard should be a replica set
- Shards hold chunks
- Chunks hold docs
  - Docs < 16MB
  - Chunks < 200MB
- Chunks defined by range on index
- Sequential batch jobs see no performance boost
- Auto balancer moves chunks transparently
- Pre-splitting recommended for batch inserts

JDK Logging Example

Morphia is mapping library using annotations to map POJOs to MongoDB JSON documents. It's very popular and may help speed up development.

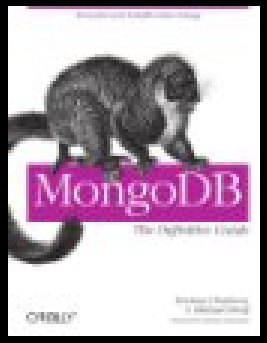
In this example however we will only use the Java driver provided by 10Gen on MongoDB.org

Questions?

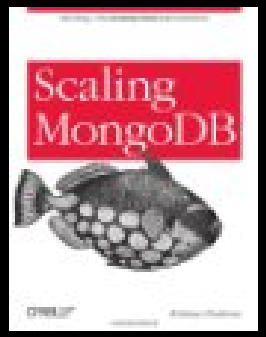
Best documentation on line:
http://www.mongodb.org/display/DOCS/Manual

OCI SETT MongoDB:
http://sett.ociweb.com/sett/settAug2011.html

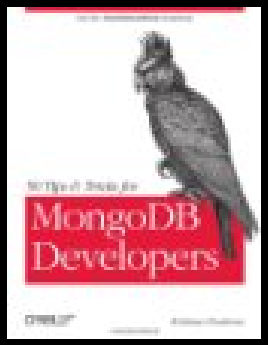
Kristina Chodorow (10gen developer):



Overview



Sharding



50 Tips and Tricks