
Couchbase

Cluster Manager

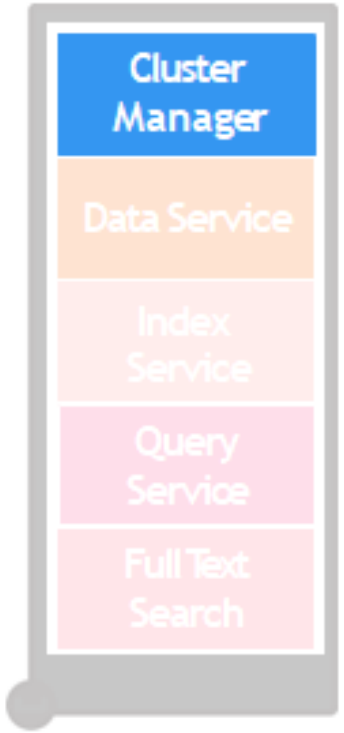
HP

The ClusterManager

The control plane of the server:

- Cluster membership
  - Status &health monitoring
- Service layout
- Data placement
  - Rebalance
  - Failover
- Authentication
- Admin APIs

Implemented in Erlang

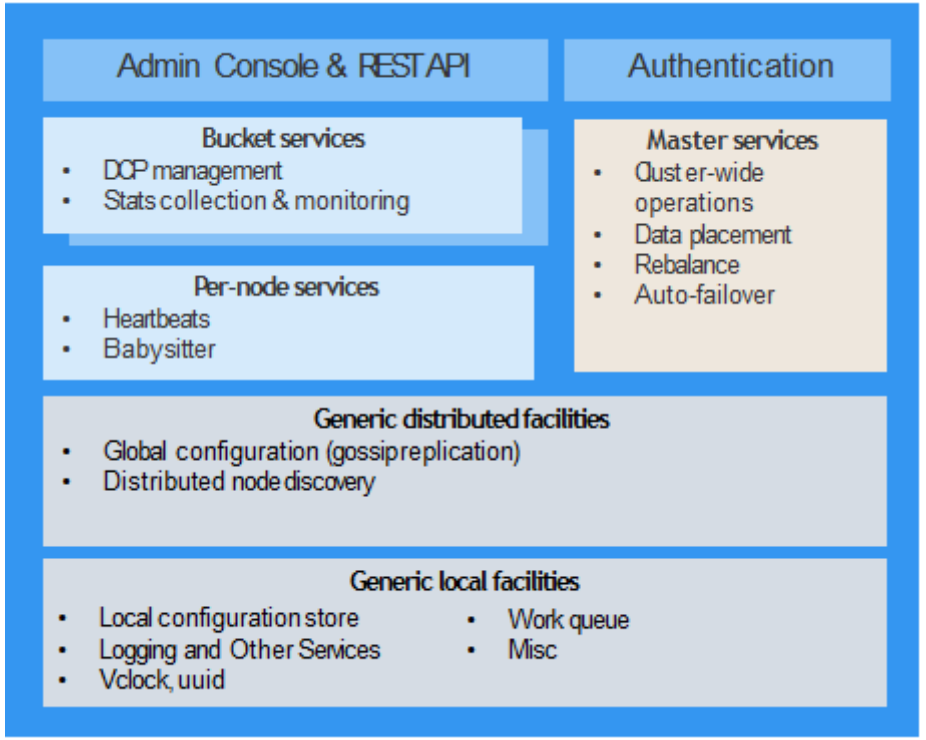


HP

The ClusterManager

- The control plane of the server:
- Cluster membership
  - Status & health monitoring
- Service layout
- Data placement
  - Rebalance
  - Failover
- Authentication
- Admin APIs

- Implemented in Erlang



Data Service

HP

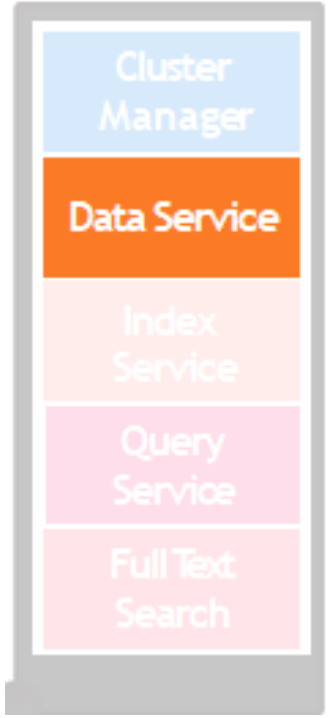
The DataService

The (low-level) data plane of the server:

- Key/Value access
- Map/Reduce Views [*]

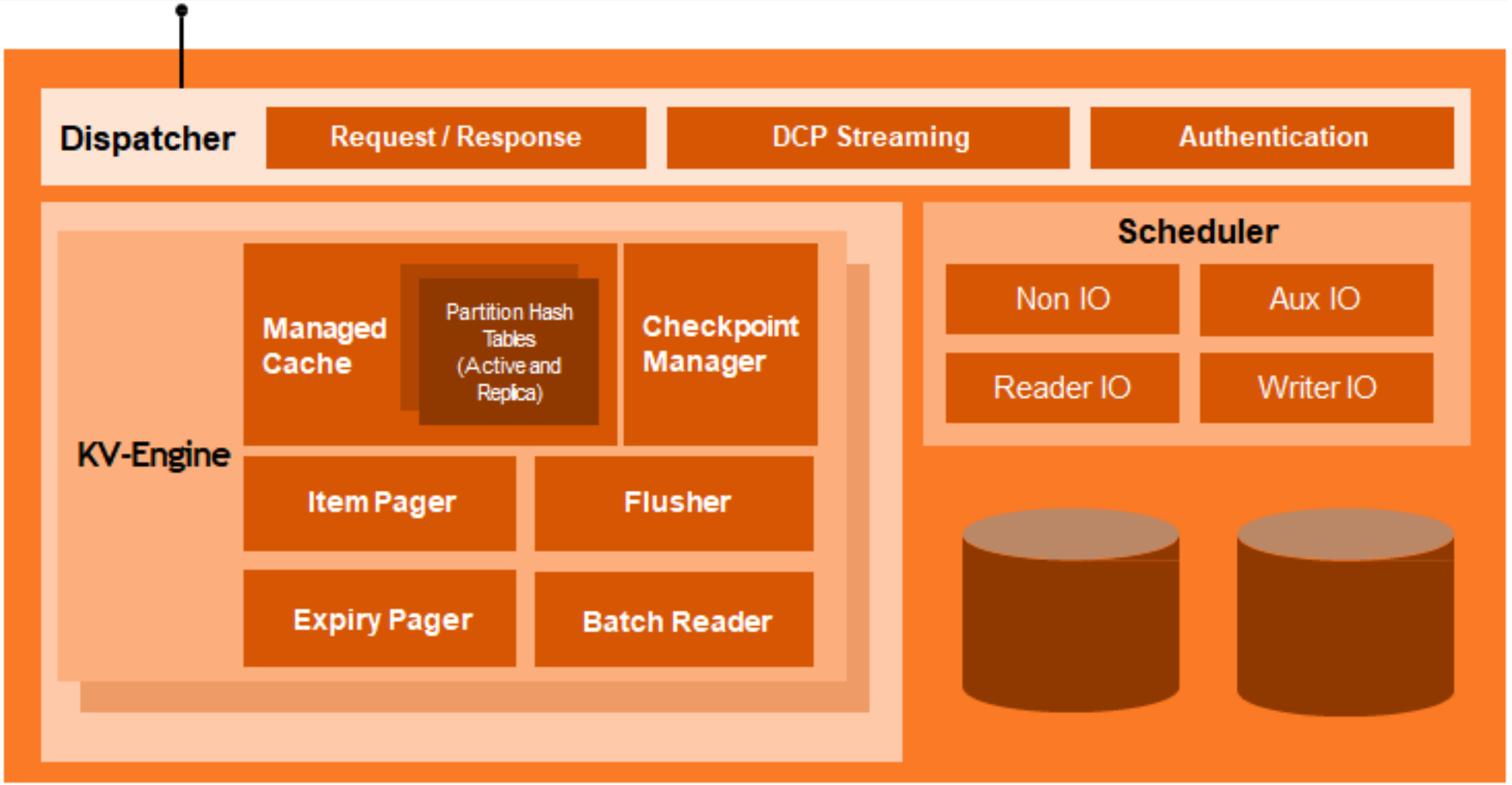
KV-Engine:

- Evolution of memcached; adding persistence, replication, enhanced data access APIs
- Asynchronous networking supports ~10K clients.
- Memory-centric architecture.
- Disk IO performed via background threads.



HP

DataService Architecture



HP

Caching Layer

- includes a built-in caching layer
- acts as a central part of the server
- automatically places items that come into the caching layer into disk queue so that it can write these items to disk.
- the entire process of managing data between the caching layer and data persistence layer is handled entirely by server

Indexing Service

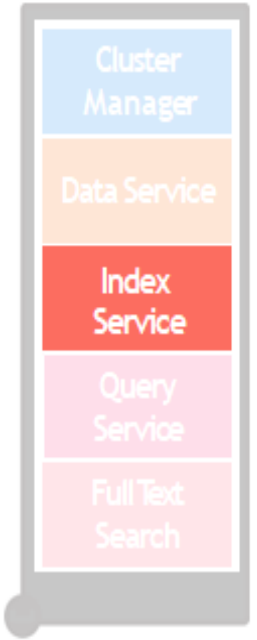
HP

Index Service

Three "indexing" services:

- Incremental Map / Reduce Views
  - Javascript map() & reduce() functions applied to all mutations
  - Supports geo-spatial views
  - Co-located with Data service
- **Global Secondary Indexes** (GSI)
- Full-Text Search – more later

GSI: efficient indexes for secondary lookups and ad-hoc query processing



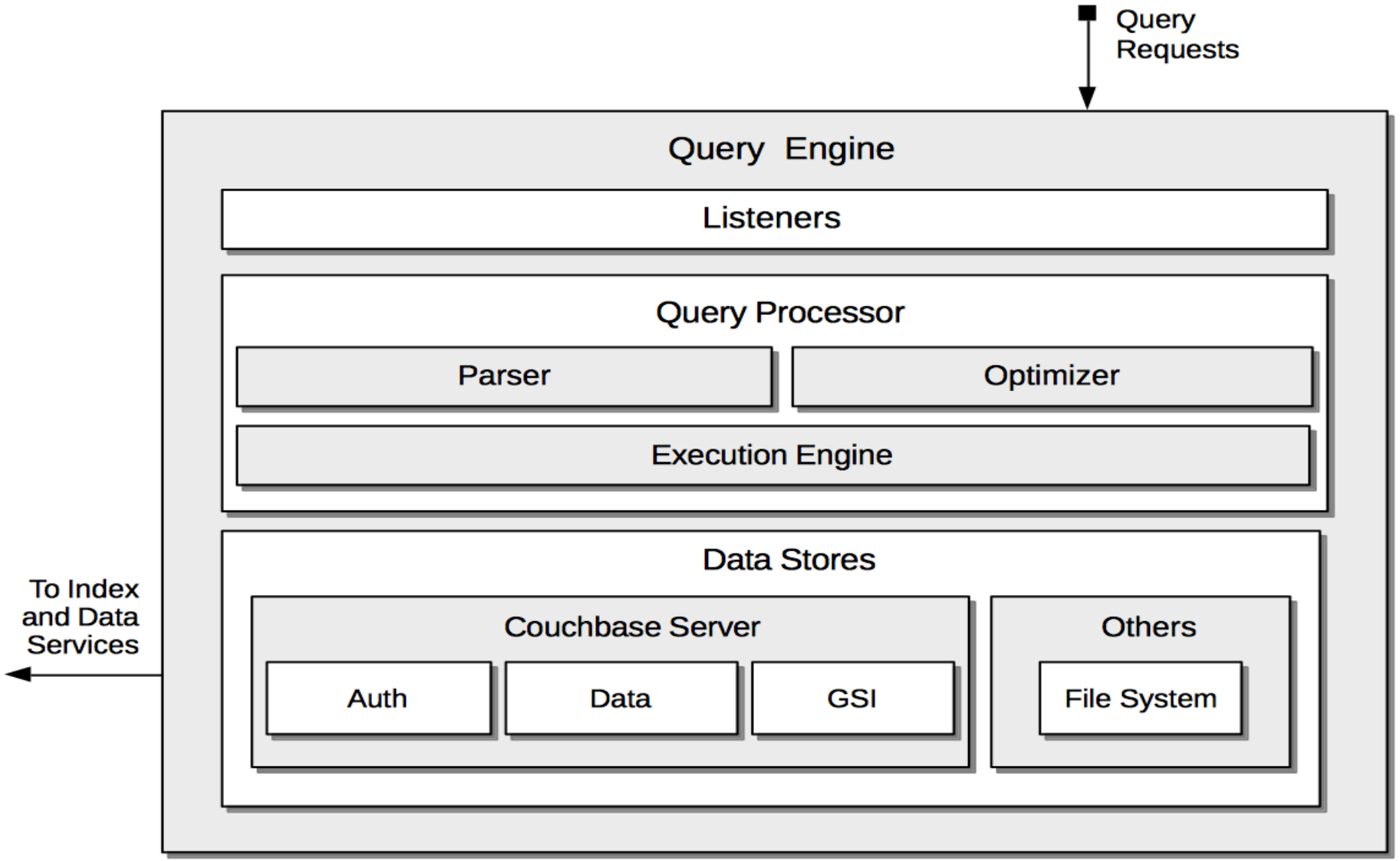
Indexing and Querying – The basics

- Define materialized views on JSON documents and then query across the data set
- Using views you can define
  - Primary indexes
  - Simple secondary indexes (most common use case)
  - Complex secondary, tertiary and composite indexes
  - Aggregations (reduction)
- Indexes are ***eventually indexed***
- Queries are ***eventually consistent***
- Built using Map/Reduce technology
  - Map and Reduce functions are written in Javascript

QueryService

HP

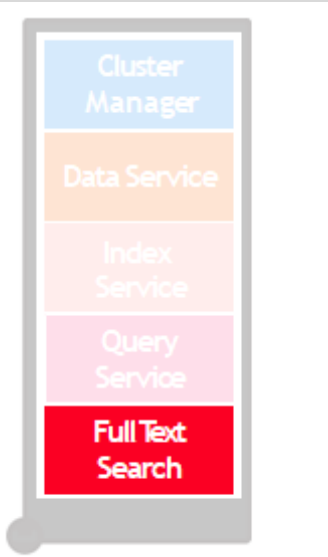
Query Service Architecture



Full TextSearch

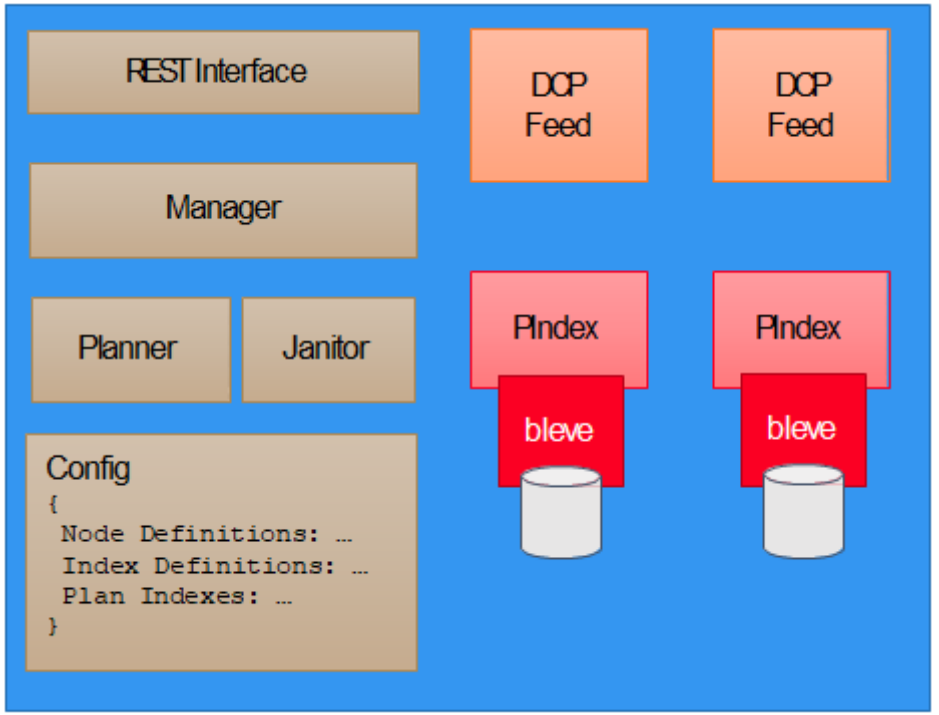
HP

Full Text Search



"Googling for your JSON documents"

- Index Fields or Documents
- Lexical Analysis and Stemming
- Flexible Query Capabilities
- Available and Scalable



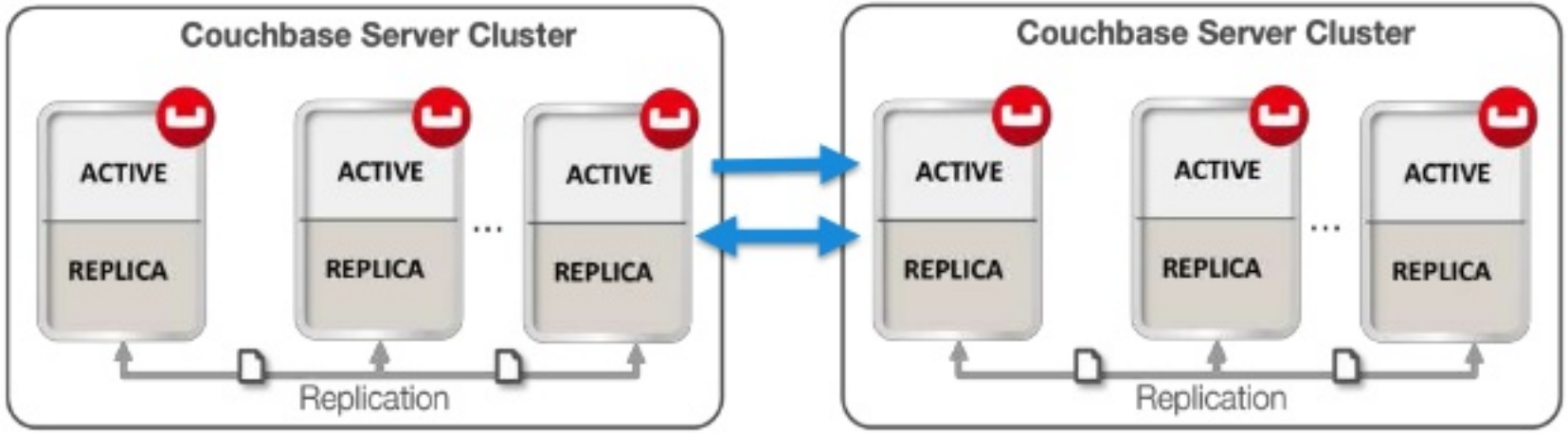
Cross Data Center Replication

HP

Cross Data Center Replication – The basics

Cross Datacenter Replication (XDCR)

- Unidirectional Replication
- Bidirectional Replication



- Hot spare / Disaster Recovery
- Development/Testing copies
- Datacenter Locality
- Multiple Active Masters

HP

Cross Data Center Replication – The basics

- Replicate your Couchbase data **across clusters**
- Clusters may be spread across geos
- Configured on a per-bucket (per-database) basis
- Supports unidirectional and bidirectional operation
- Application can read and write from both clusters
  - Active – Active replication
- Replication throughput scales out linearly
- Different from intra-cluster replication

HP

Couchbase and Traditional RDMS

Couchbase Server	(RDBMS)
Rapidly scalable to millions of users.	Scalable to thousands of users.
Data can be structured, semi-structured, and unstructured	Data must be normalized.
Data can be flexibly stored as JSON documents or binary data. No need to predefine data types.	Data types must be predefined for columns.
Data stored as key-document pairs; well suited for applications which handle rapidly growing lists of elements.	Data stored in tables with fixed relations between tables.
Asynchronous operations and optimistic concurrency enable applications designed for high throughput.	Strict enforcement of data integrity and normalization, with the tradeoff of lower performance and slower response times.

Couchbase Server Startup and Shutdown & Testing Nodes - 30 Minutes(D)