Cassandra Essentials Tutorial Series

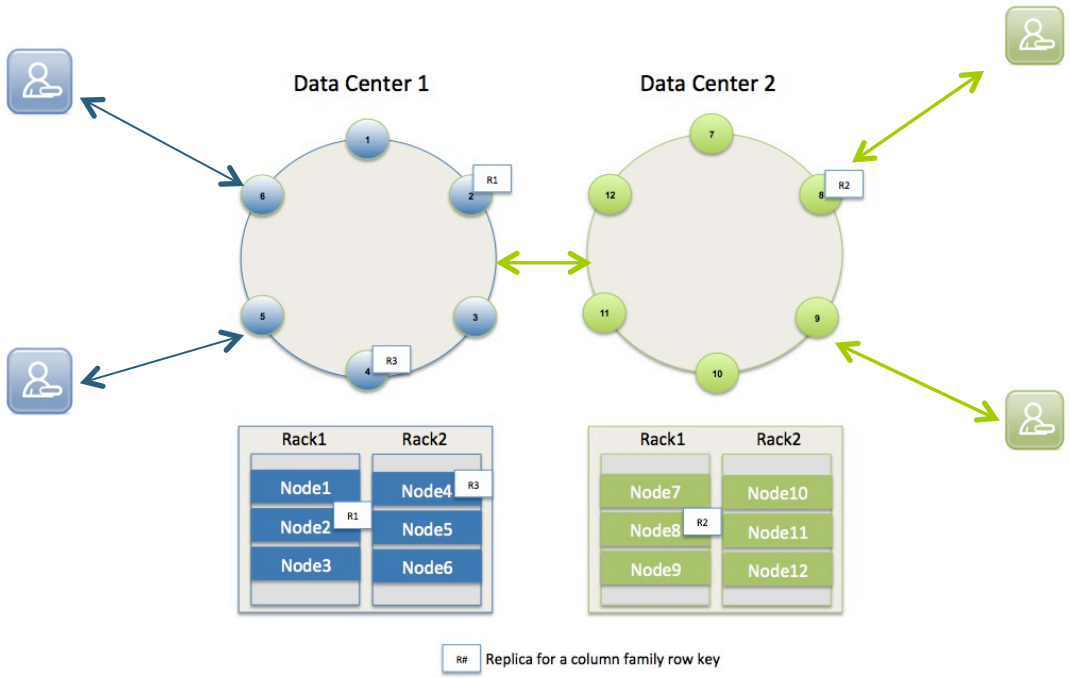
Understanding Data Consistency in Apache Cassandra

Agenda

- Overview of reading/writing data in Cassandra
- Details on how Cassandra writes data
- Review of the CAP theorem
- Tunable data consistency
- Choosing a data consistency strategy for writes
- Choosing a data consistency strategy for reads
- CQL examples of data consistency
- Where to get Cassandra

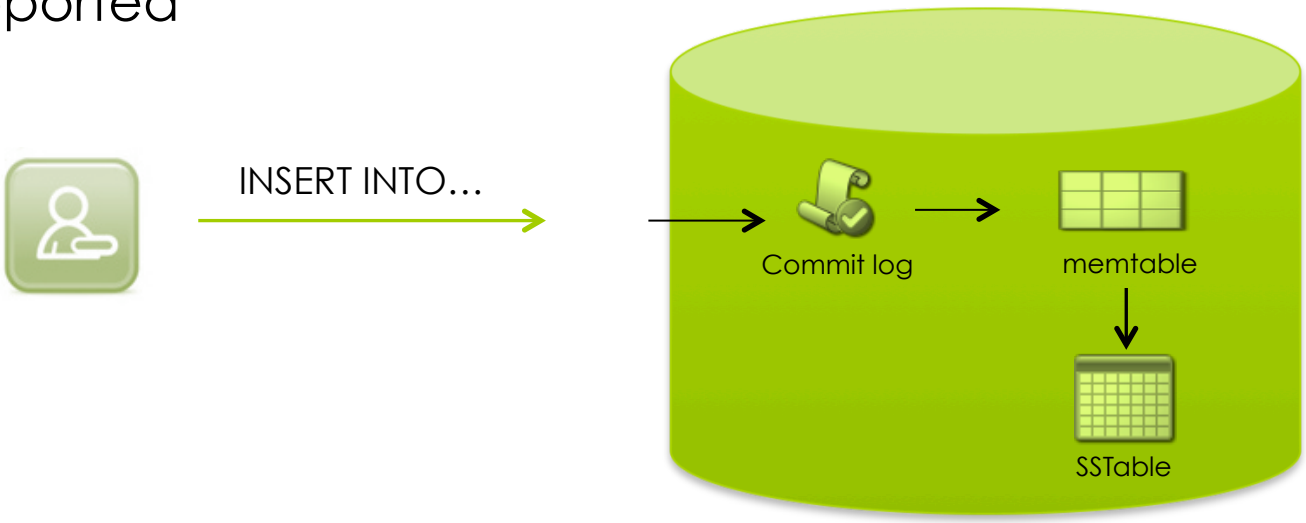
Reading and Writing in Cassandra

Cassandra is a peer-to-peer, read/write anywhere architecture, so any user can connect to any node in any data center and read/write the data they need, with all writes being partitioned and replicated for them automatically throughout the cluster.



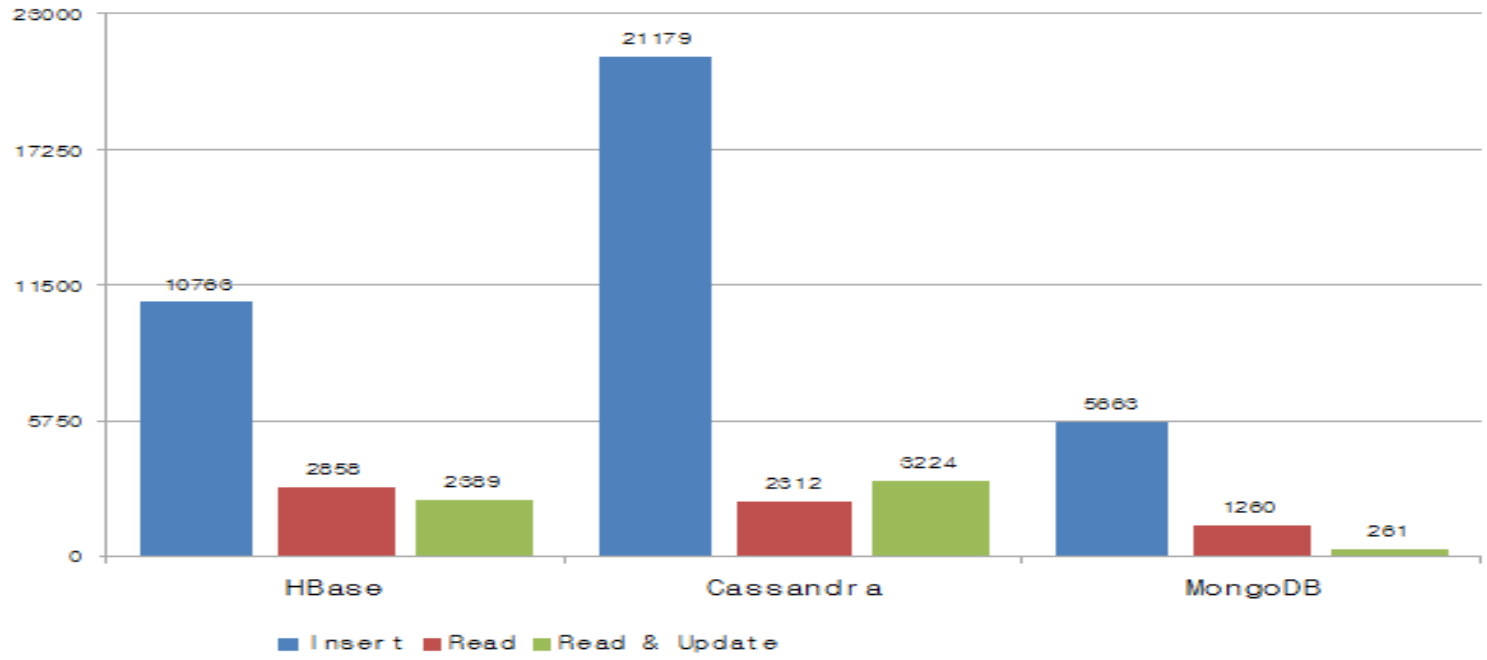
Writes in Cassandra

- Data is first written to a commit log for durability
- Then written to a memtable in memory
- Once the memtable becomes full, it is flushed to an SSTable (sorted strings table)
- Writes are atomic at the row level; all columns are written or updated, or none are. RDBMS-styled transactions are not supported



Cassandra is known for being the fastest database in the industry where write operations are concerned.

Writes in Cassandra vs. Other Databases



Cassandra is up to:

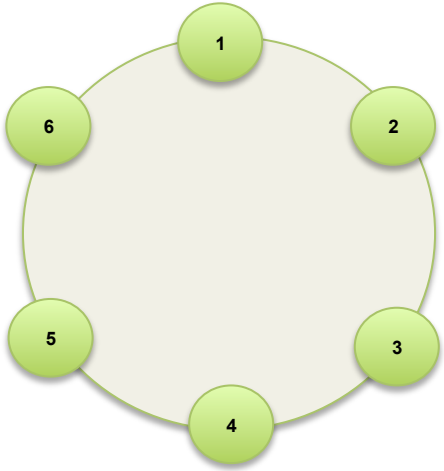
4x better in writes!
2x better in reads!
12x better in reads/updates!

Sept, 2011: http://blog.cubrid.org/dev-platform/nosql-benchmarking/

Review of the CAP Theorem

Tunable Data Consistency

- Choose between strong and eventual consistency (All to any node responding) depending on the need
- Can be done on a per-operation basis, and for both reads and writes
- Handles Multi-data center operations



Writes

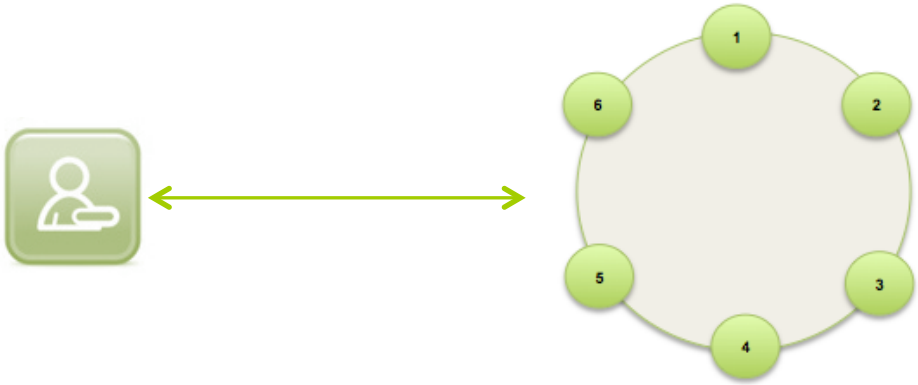
- Any
- One
- Quorum
- Local_Quorum
- Each_Quorum
- All

Reads

- One
- Quorum
- Local_Quorum
- Each_Quorum
- All

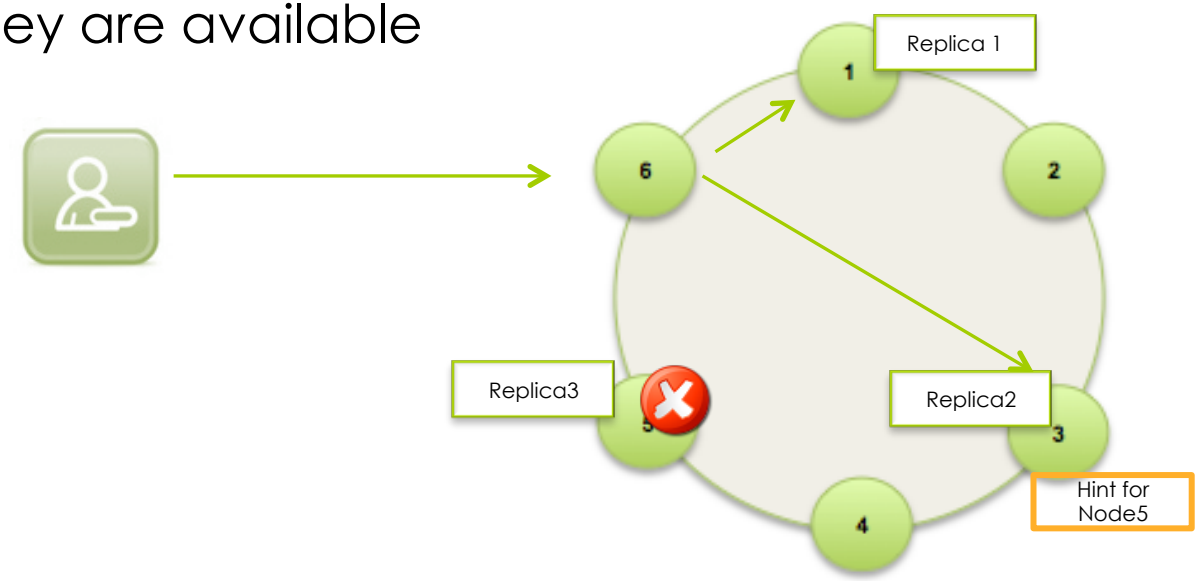
Selecting a Strategy for Writes

- **Any** – a write must succeed on any available node
- **One** – a write must succeed on any node responsible for that row (either primary or replica)
- **Quorum** – a write must succeed on a quorum of replica nodes (determined by (`replication_factor` /2)+ 1
- **Local_Quorum** - a write must succeed on a quorum of replica nodes in the same data center as the coordinator node
- **Each_Quorum** - a write must succeed on a quorum of replica nodes in all data centers
- **All** – a write must succeed on all replica nodes for a row key



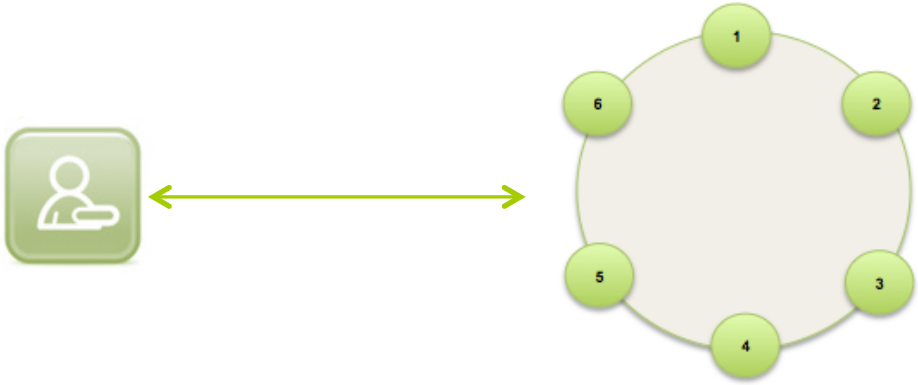
Hinted Handoffs

- Cassandra attempts to write a row to all replicas for that row
- If all replica nodes are not available, a hint is stored on one node to update any downed nodes with the row once they are available again
- If no replica nodes are available for a row, the use of the ANY consistency level will instruct the coordinator node to store a hint and the row data, which it passes to the replica nodes when they are available



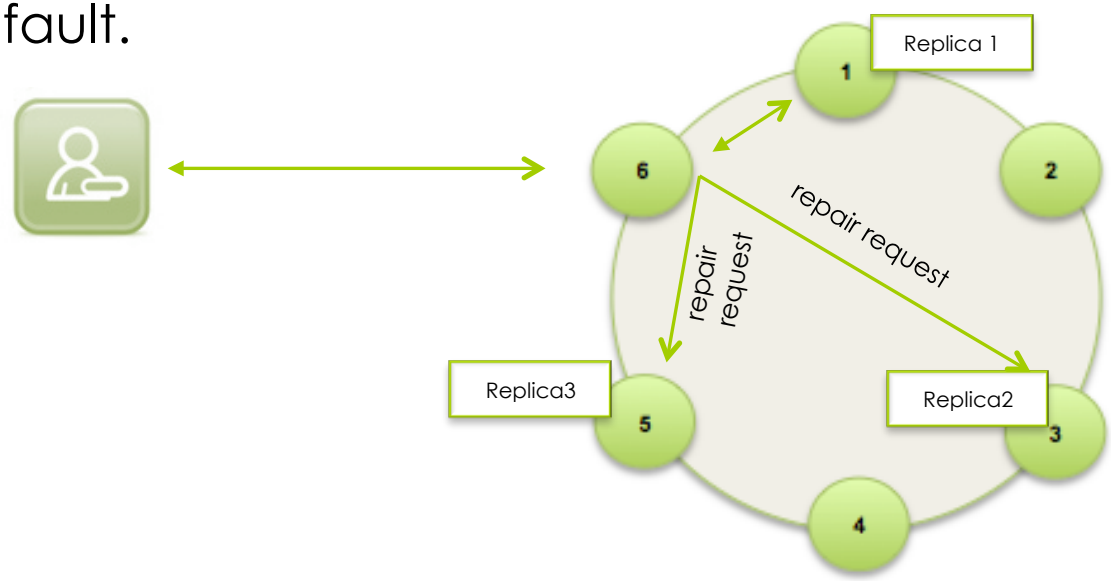
Selecting a Strategy for Reads

- **One** – reads from the closest node holding the data
- **Quorum** – returns a result from a quorum of servers with the most recent timestamp for the data
- **Local_Quorum** - returns a result from a quorum of servers with the most recent timestamp for the data in the same data center as the coordinator node
- **Each_Quorum** - returns a result from a quorum of servers with the most recent timestamp in all data centers
- **All** – returns a result from all replica nodes for a row key



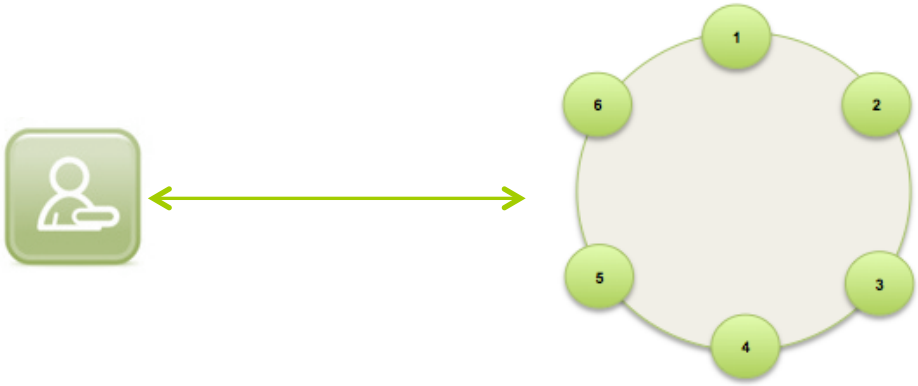
Read Repair

- Cassandra ensures that frequently-read data remains consistent
- When a read is done, the coordinator node compares the data from all the remaining replicas that own the row in the background, and if they are inconsistent, issues writes to the out-of-date replicas to update the row to reflect the most recently written values.
- Read repair can be configured per column family and is enabled by default.



CQL Examples

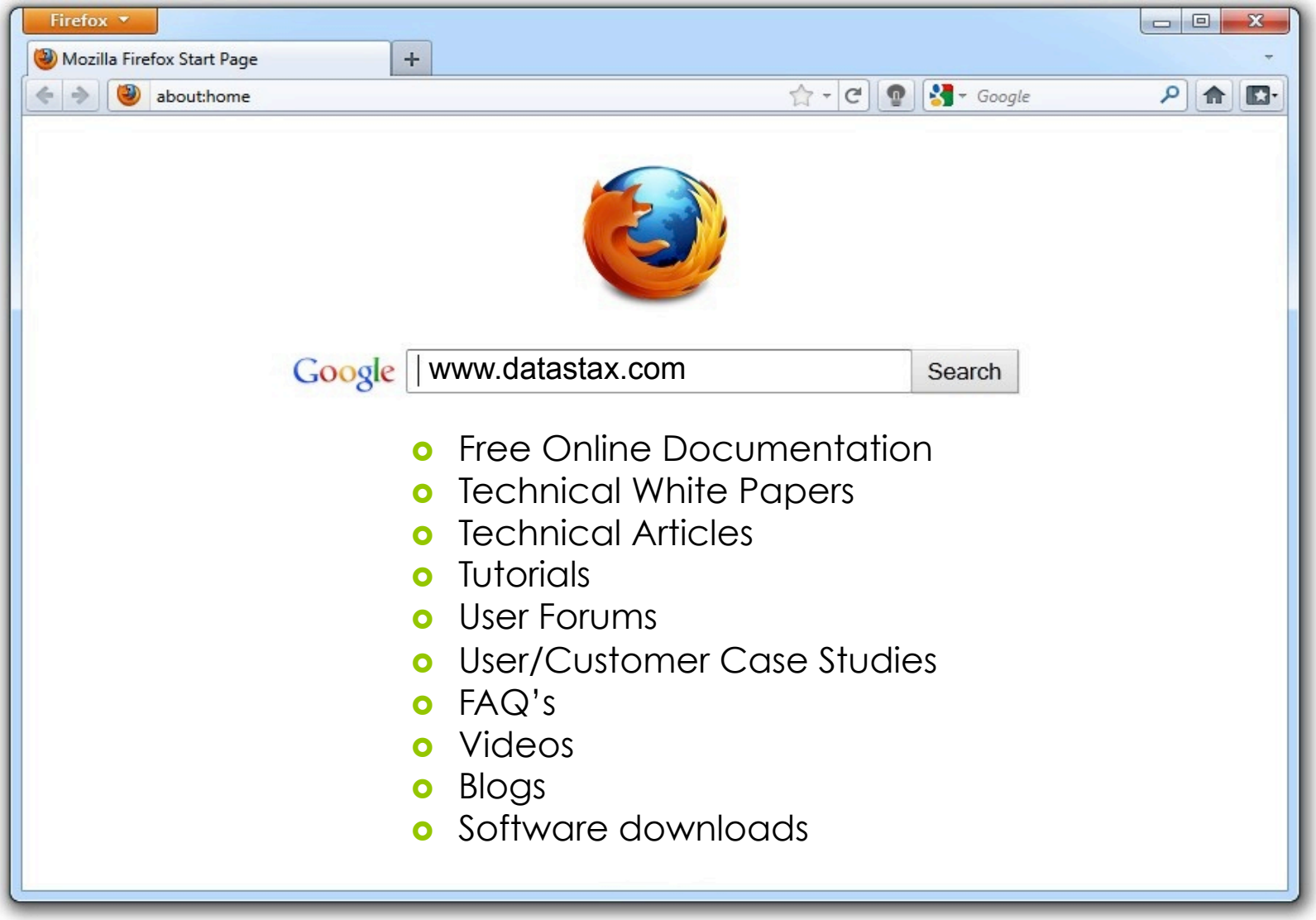
```
SELECT total_purchases FROM SALES
USING  CONSISTENCY QUORUM
WHERE  customer_id = 5

UPDATE SALES
USING  CONSISTENCY ONE
SET    total_purchases = 500000
WHERE  customer_id = 4
```



Where to get Cassandra?

- Go to www.datastax.com
- DataStax makes free smart start installers available for Cassandra that include:
  - The most up-to-date Cassandra version that is production quality
  - A version of DataStax OpsCenter, which is a visual, browser-based management tool for managing and monitoring Cassandra
  - Drivers and connectors for popular development languages
  - Same database and application
  - Automatic configuration assistance for ensuring optimal performance and setup for either stand-alone or cluster implementations
  - Getting Started Guide

Where Can I Learn More?



- Free Online Documentation
- Technical White Papers
- Technical Articles
- Tutorials
- User Forums
- User/Customer Case Studies
- FAQ's
- Videos
- Blogs
- Software downloads

Cassandra Essentials Tutorial Series

Understanding Data Partitioning and Replication in Apache Cassandra

Thanks!