Oracle NoSQL Database

Compared to CouchDB

Overview

- Oracle NoSQL Database is licensed under AGPL while CouchDB is Apache 2.0 licensed.
- Oracle NoSQL Database is in many respects, as a NoSQL Database implementation leveraging BerkeleyDB in its storage layer, a commercialization of the early NoSQL implementations which lead to the adoption of this category of technology. Several of the earliest NoSQL solutions were based on BerkeleyDB and some are still to this day e.g. LinkedIn's Voldemort. The Oracle NoSQL Database is a Java based key-value store implementation that supports a value abstraction layer currently implementing Binary and JSON types. Its key structure is designed in such a way as to facilitate large scale distribution and storage locality with range based search and retrieval. The implementation uniquely supports built in cluster load balancing and a full range of transaction semantics from ACID to relaxed eventually consistent. In addition, the technology is integrated with important open source technologies like Hadoop / MapReduce, an increasing number of Oracle software solutions and tools and can be found on Oracle Engineered Systems.
- CouchDB is a key-value store that supports document storage.

Comparison

The table below gives a high level comparison of Oracle NoSQL Database and CouchDB features/capabilities. Low level details are found in links to Oracle and CouchDB online documentation.

Feature/Capability	Oracle NoSQL Database	CouchDB
Data Model	Oracle NoSQL Database has a flexible key-value data model leveraging a value abstraction layer. The value abstractions supported at this time are Binary and JSON(Avro). A table-structure value abstraction is coming soon • Record Design Considerations • Avro Schemas	CouchDB's data format is JSON stored as documents (self-contained records with no intrinsic relationships), grouped into "database" namespaces. • Document API
Storage Model	Oracle NoSQL Database storage model is a write ahead logging implementation proven in millions of BerkeleyDB deployments. It's an append only implementation that enables efficient write throughput with background compaction for space reclamation. Write operation durability can be controlled by the user to allow multi-memory write operations without fsync or with fully durable disk sync. Data is partitioned into a fix space that has logical overlays. So, data in partitions can move between logical shard representations, but must be moved at the granularity of these partitions. • BDB Storage - NoSQL before NoSQL was	CouchDB stores data to disk by "append-only" files. As the files continue to grow, they require occasional compaction. • Indexes and File

	cool • The evolution of BerkeleyDB	
Data Access and APIs	Oracle NoSQL Database has client library API's for Java and C. In the works are a Command Line Interface and Javascript API. • Client APIs	CouchDB provides an HTTP API for both data access and administration. • Document API • View API • DB API The CouchDB community supports many client libraries. • Client-Libraries
Query Types and Query-ability	Oracle NoSQL Database provides key access methods (put, get, delete) including multi-key variations with large result set streaming support. The database can also be accessed using SQL as an external table from within a relational database. It is integrated with and can participate in MapReduce operations from a Hadoop environment. • Searching in Oracle NoSQL • External Table Support • NoSQL and MapReduce • Using Range Queries	CouchDB is generally queried by direct ID lookups, or by creating MapReduce "views" that CouchDB runs to create a queryable index for querying by or computing other attributes. In addition, the ChangesAPI shows documents in the order they were last modified. Finally, there exist some community plugins to expand CouchDB's queryability, such as the CouchDB-Lucene full-text search plugin. • Views • Changes Notifications • Lucene Plugin
Data Versioning and Consistency	Oracle NoSQL Database provides control at the	CouchDB replicates newer document versions between nodes, making it an

	operation level for consistency and durability. Each operation can be fully ACID, flushing and syncing all data to disk before taking quorum on the operation to allowing a fire and forget into local or remote memory. Read consistency is obtained thru quorum control spanning the range of requiring all holders of a copy of data to agree to just getting the result from a first responder. This provides the ultimate control for the developer of both transactional and eventually consistent applications. Flexible Consistency options	eventually consistent system. CouchDB uses Multi-Version Concurrency Control (MVCC) to avoid locking the database file during writes. Conflicts are left to the application to resolve at write time. Older document versions (called revisions) may be lost when the append-only database file is compacted. • Eventual Consistency
Concurrency	Oracle NoSQL Database concurrency is controlled thru replication groups with an elected master. Reads can be serviced from any node in a replication group and writes are performed at the currently elected master, then replication chained to the replicas in the group. Read consistency is tied to concurrency, controlled by quorum, version, timestamp, all. Durability Guarantees	Because of CouchDB's append-only value mutation, individual instances will not lock. When distributed, CouchDB won't allow updating similarly keyed document without a preceding version number, and conflicts must be manually resolved before concluding a write. • No Locking • Conflict Management
Replication	Oracle NoSQL Database supports replication for both availability and scalability. It	CouchDB incrementally replicates document changes between nodes. It can be deployed with master/master

	uses a consistent hashing algorithm over a fixed, highly granular, partition definition. Partitions are replicated in groups according to latency demands of the application, configured by a replication factor. • Replication configuration There is a topology aware driver that is linked with the client application. Writes use the driver to hash inserts to the currently elected master and then a cascading replication occurs to the replicas belonging to the replication group where that master resides. How many data replications must occur and whether or not those replications are to memory space or disk for the respective replica can be configured on a per operation basis. • Topologies	or master/slave replication. Replication can be finely controlled by way of replication filters. • Replication
Scaling Out and In	Oracle NoSQL Database scales out by redistribution of data partitions to newly added hardware resources. When new hardware is added to the system, an administrator, via a browser based console or CLI, can issue a request to rebalance the cluster. The	Out of the box, CouchDB is focused on a master-master replication of values (using MVCC to help with conflict resolution). There are external projects that help manage a CouchDB cluster, such as BigCouch (also Apache 2.0 licensed), that shards values across multiple nodes.

	administrator has the option of just letting it go or throttling or running during certain windows of time, pausing the process, etc. • Managing Topology Changes	• BigCouch • Sharding (on Wikipedia)
Multi-Datacenter Replication and Awareness	Oracle NoSQL Database supports DataCenters thru a non-electable replication group strategy. Read requests use nodes locally due to latency awareness in the client driver. Write availability is achieved in a local quorum though replicating to non-electable nodes in other data centers. This allows failures in a given data center to have no impact on read availability of the cluster as a whole, just possibly some reduced latency. Writes will always be performed at the currently elected master.	CouchDB can be configured to run in multiple datacenters. Robust awareness will generally require a third part solution, or by developing replication filters. • Filtered Replication • The Split Brain
Graphical Monitoring/Admin Console	Oracle NoSQL Database provides proprietary, SNMP and JMX based protocols for monitorability of the cluster. The proprietary protocols are support thru both browser based and CLI interfaces. SNMP and JMX facilitate integration into monitoring systems like BMC and Ganglia. • Visual Admin Console • Standardized	CouchDB ships with a graphical interface called Futon.

	Monitoring Protocols • Command Line Admin	