The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Agenda

• Oracle NoSQL Database Overview
• Release 2.0 Features
• NoSQL DB Use Cases
Big Data Lifecycle

**Acquire** all available data
# Big Data Storage Choices

## Hadoop Distributed File System (HDFS)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Oracle NoSQL Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>File System</td>
<td>Database</td>
</tr>
<tr>
<td>Parallel scanning</td>
<td>Indexed storage</td>
</tr>
<tr>
<td>No inherent structure</td>
<td>Simple data structure</td>
</tr>
<tr>
<td>High volume writes</td>
<td>High volume random reads and writes</td>
</tr>
<tr>
<td>Batch Oriented</td>
<td>Real-Time, “Last Mile to the customer” applications</td>
</tr>
</tbody>
</table>
Where does Oracle NoSQL DB fit?

- Best database for the “last mile to the customer“ applications
- In the Big Data Acquisition phase
- Complements Hadoop/HDFS (batch), Oracle Database
- Integrates with OEP, RDF, others
Typical Big Data/NoSQL Solution Architecture

Real Time Access

Big Data Acquisition

Batch Processing

Application
NoSQL DB Driver

HDFS, Hadoop, CDH

Map Reduce
ORCH - Stats
Map Reduce
Pig - Sessionize
Map Reduce
Hive - Activities

Oracle Big Data Appliance

Endeca Information Discovery

Oracle Business Intelligence EE

Oracle Exalytics

ORCH - Stats

Pig - Sessionize

Hive - Activities

Oracle Advanced Analytics

Oracle Exadata

OLH, ODC, ODI, External Tables

ORE, OEP, Endeca

Typical Big Data/NoSQL Solution Architecture
Sample of Big Data Use Cases Today

**AUTOMOTIVE**
- Auto sensors reporting location, problems

**COMMUNICATIONS**
- Location-based advertising

**CONSUMER PACKAGED GOODS**
- Sentiment analysis of what’s hot, problems

**FINANCIAL SERVICES**
- Risk & portfolio analysis
- New products

**EDUCATION & RESEARCH**
- Experiment sensor analysis

**HIGH TECHNOLOGY / INDUSTRIAL MFG.**
- Mfg quality warranty analysis

**LIFE SCIENCES**
- Clinical trials Genomics

**MEDIA/ENTERTAINMENT**
- Viewers / advertising effectiveness

**ON-LINE SERVICES / SOCIAL MEDIA**
- People & career matching
- Web-site optimization

**OIL & GAS**
- Drilling exploration sensor analysis

**RETAIL**
- Consumer sentiment optimization

**TRAVEL & TRANSPORTATION**
- Sensor analysis for optimal traffic flows
- Customer sentiment

**UTILITIES**
- Smart Meter analysis for network capacity,

**FINANCIAL SERVICES**
- Risk & portfolio analysis
- New products

**HIGH TECHNOLOGY / INDUSTRIAL MFG.**
- Mfg quality warranty analysis

**LIFE SCIENCES**
- Clinical trials Genomics

**MEDIA/ENTERTAINMENT**
- Viewers / advertising effectiveness

**ON-LINE SERVICES / SOCIAL MEDIA**
- People & career matching
- Web-site optimization

**OIL & GAS**
- Drilling exploration sensor analysis

**RETAIL**
- Consumer sentiment optimization

**TRAVEL & TRANSPORTATION**
- Sensor analysis for optimal traffic flows
- Customer sentiment

**UTILITIES**
- Smart Meter analysis for network capacity,

**LAW ENFORCEMENT & DEFENSE**
- Threat analysis - social media monitoring, photo analysis

**Challenged by:** Data Volume, Velocity, Variety

*Oracle NoSQL Database is typically a component of a Big Data Solution*
Oracle NoSQL Database Use Cases

Use Cases

- High-throughput event processing
- Customer profile management
- Click-through data processing
- Sensor & statistics data capture
- Social networks
- Personalization
- Mobile application backend infrastructure
- Authentication & Content management
- Archiving

SIMPLE QUERIES

DYNAMIC SCHEMA

HIGH VOLUME

REAL TIME DATA ACCESS

“Last mile” problems
Oracle NoSQL Database
Scalable, Highly Available, Key-Value Database

Features

▪ Simple Key-Value Data Model
▪ Horizontally Scalable
▪ Highly Available
▪ Elastic Configuration
▪ Simple administration
▪ Transparent load balancing
▪ Commercial grade software and support
Oracle NoSQL Database
Technical Summary

• Scalable
  – Dynamic data partitioning and distribution
  – Automatic Query Load Balancing
  – High Performance and Throughput

• Highly Available
  – One or more replicas
  – No single point of failure

• Optimized, intelligent database driver

• Simple Key-Value data model

• C and Java APIs
• JSON Schemas
• ACID Transactions
• Easy to install and manage

• Requirements
  – Java SE 6 (JDK 1.6.0 u25)+
  – Solaris 10 or Linux
# Oracle NoSQL Database

## Key Features

<table>
<thead>
<tr>
<th>NoSQL DB Feature</th>
<th>User Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed storage and queries</td>
<td>Scalability, Performance</td>
</tr>
<tr>
<td>Intelligent database driver</td>
<td>Performance, Scalability, Reliability</td>
</tr>
<tr>
<td>No Single Point of Failure</td>
<td>Reliability, 99.999% availability</td>
</tr>
<tr>
<td>Simple Key-Value storage</td>
<td>Performance, flexibility, ease of use</td>
</tr>
<tr>
<td>ACID transactions</td>
<td>Reliability, data integrity, ease of use</td>
</tr>
<tr>
<td>Simple Administration</td>
<td>Low OPEX, ease of use</td>
</tr>
</tbody>
</table>
Simple Data Model

Key-value pairs

- Simple data model – key-value pair (major+minor-key paradigm)
- Simple operations – read/insert/update/delete, RMW support
- Scope of transaction – records within a major key, single API call
- Unordered scan of all data (non-transactional)

**Major key:**
- Strings
  - userid

**Minor key:**
- subscriptions
- address

**Value:**
- expiration date
- phone #
- email id
Key Structure Design - Examples

**Social Network**
- **Major Key**: /john smith/
- **Minor Key**: tweets, birthday
- **Value**: tweet001, tweet002, 1970-1-1

**Email**
- **Major Key**: /steve.jobs/
- **Minor Key**: inbox, sent
- **Value**: thread001, thread002, thread002

**CDR**
- **Major Key**: /calling/
- **Minor Key**: today, yesterday
- **Value**: CDR001, CDR002, CDR003
Simple Data Model

ACID Transactions

• ACID transactions by default

• Transaction Scope
  – Single API call
  – All records must have the same major key
  – Support for multiple operations within a transaction

• Can be relaxed for increased performance on a per-operation basis
Simple Data Model

ACID Transactions – Configurability

• Configurable Durability Policy

  write
  - Memory
  - FS Buffer
  - Disk

  +

  HA ack
  - None
  - Majority
  - All

  Fastest
  Most Durable

• Configurable Consistency Policy

  guarantees
  - Can Read Stale Data
  - Data is recent as of given time
  - Operating on Known or later version
  - Operating on most recent version

  Fastest
  Most Consistent

  consistency
  - None
  - Time-Based
  - Version-Based
  - Absolute
Easy to use APIs

C or Java

• Simple CRUD operations
  • Conditionalized to reduce client/server round trips

• Iteration operations
  • Returns multiple results in a single API call
  • Get full records or just the keys

• Multi and Sub-Key operations
  • Multiple operations and/or Multiple records in a single transaction
Scalability and Availability

Application Perspective
Scalability and Availability

- Replicated Application Servers
- Driver is linked into each Application
- Storage Nodes kept current via replication (Berkeley DB Java Edition HA)
- Storage Nodes across Data Centers
- Automatic SN failure handling
  - Graceful degradation
  - Automatic recovery

➢ No Single Point of Failure
Oracle NoSQL Database Differentiation

Integrates seamlessly with Oracle Stack (ODI, CEP, OLH)

**Commercial Grade Software and Support**
- General Purpose
- Reliable – Based on proven Berkeley DB JE HA
- Easy to Install & Configure

**Scalability and Availability**
- Intelligent Oracle NoSQL DB Driver
  - Evenly distributes Data
  - Sends operation to fastest node
  - Bounded network hops for all operations
- Automatic replication and failover

**Simple Data Model**
- Simple Major + Minor Key-Value data structure
- ACID transactions
- Configurable consistency and durability

**Simple Administration**
- Web-based Console and CLI commands
- Manages and Monitors:
  - Topology
  - Load
  - Performance
  - Events
  - Alerts

Integrates seamlessly with Oracle Stack (ODI, CEP, OLH)
YCSB Benchmark Results

- 1.25M ops/sec
- 2 billion records
- 2 TB of data
- 95% read, 5% update
- Low latency
- High Scalability
Agenda

• Oracle NoSQL Database Overview
• Release 2.0 Features
• NoSQL DB Use Cases
What’s New?

Release 2 Feature Summary

R2 Features

Scalability & Manageability
- Elasticity
- Rebalancing
- Smart Topology

New APIs
- JSON schemas
- C-API
- Large Object Support

Integration & Monitoring
- External Tables
- Oracle Event Processing
- RDF Adapter
- SNMP/JMX
Configurable Smart Topology

• Storage nodes have indication of “capacity”
• System may allocate multiple replication nodes per storage node
• Intelligent node balancing
• Efficient use of system resources
• Reduce time for recovery of disk media failure
JSON Data Format

Avro based Serialization/Deserialization for the Value (KV)

• Why Avro?
  – Compact, highly efficient serialization
  – Synergy with Hadoop

• Schema
  – DDL allows schema creation through Avro JSON definition
  – Supports serialization from/to JSON strings

• Schema evolution
  – Easy to use mechanism for schema evolution
  – Schema versions can be opaque to readers
Support for Large Objects

• Efficient storage and retrieval of large objects
• Client side streaming interface for low memory consumption
• Server side splitting and distribution of object chunks across nodes for better read/write latency
Enhanced System Monitoring

- **SNMP (Enterprise Edition Only)**
  - MIBs in SNMPv2 format
  - Storage node as SNMP agent – Proxies to admin and replication nodes
  - Performance metrics – Via polling
  - Service change events – Via traps/notifications (UDP) or polling
  - Notifications (Traps) – Latency and throughput limits

- **Java JMX**
  - Performance metrics – Via subscriptions
  - Service changes events – Via subscriptions
  - Notifications – Via subscriptions
External Table Support

• Available with Oracle NoSQL DB Enterprise Edition
• Oracle Database SQL access to NoSQL Database data

• Steps:
  1. Create NoSQL DB table formatter (use sample template)
  2. Define External Table in SQL
  3. Define Configuration file (use sample XML template)
  4. Use NoSQL Database Publish utility
  5. Use SQL to access NoSQL data
Integration with other Oracle Products

• Available in Oracle NoSQL Database Enterprise Edition only!
• Oracle Event Processing (OEP)
  – OEP NoSQL Database cartridge reads data from NoSQL Database
  – Flexible Java serialization utilized for values
• Oracle RDF
  – Jena Adapter for NoSQL Database to be released by the RDF product team
  – Enables RDF data storage and SPARQL queries on NoSQL DB
Agenda

• Oracle NoSQL Database Overview
• Release 2.0 Features
• NoSQL DB Use Cases
High Profile Oracle Key-Value customers

- Amazon.com
- NASDAQ
- LinkedIn
- Cisco
- RSA Security
- AT&T
- TIBCO
- Verizon Wireless
Oracle NoSQL Database

Success Stories

• ENGINEON provides PaaS for deploying applications over the cloud
  – Oracle NoSQL Database exposed as a service through their PaaS web application server infrastructure

• DBA, Oracle Platinum Partner, built an online gaming application for their customer (Passoker) using Oracle NoSQL Database

• CGBU launches Elastic Charging Engine based on NoSQL DB as part of Billing application
Passoker Sales Win

Global Sports Betting application

• Challenges
  – Provide a platform for real-time sports data feeds
  – Deploy scalable, efficient, and highly technology for incoming data stream
  – Establish a technology to receive XML files and rapidly process them for relevance
  – Reduce development times and simplify day-to-day administration
  – Scale-out the platform as demand grows and players take up gaming in greater numbers

• Benefits of Oracle NoSQL Database
  – Reduced development and implementation time by 75%
  – Simplified day-to-day administration
  – Simplified rapid correlation between events and gaming
  – Improved operational efficiency due to ongoing commercial support
  – Eliminated daily maintenance related to single node point-of-failure
  – Enabled more application development and integration, better testing, shorter time to market
NoSQL Example: On-line Ads and Content

Goal: Optimize Internet Advertising to Increase Revenues

Scenario

- User clicks on a web page
- Cookie used to look up user profile
- Appropriate ad placed on page

Key Features

- Fast response times and high throughput
- High availability and scalability
- Simple data, flexible data model
- Connectivity to Big Data Analytics and Discovery
NoSQL Example: Remote Patient Monitoring
Goal: Better Patient Care at Lower Cost

Scenario
- Patient uses multiple devices at home
- Medical data periodically sent to NoSQL DB
- App monitors and alerts patient state
- Alerts sent to medical or emergency personnel, recorded in profile

Key Features
- High performance and high availability
- High throughput event capture
- Huge volumes of data
- Simple data, flexible data model
Example: Credit Card Transaction Authorization

Goal: Better, Faster Fraud Detection and Response

Scenario
- Credit card swipe at point of sale (PoS)
- Look up customer profile (interactive)
- Authorization systems determines approval
- Approval sent to PoS and recorded

Key Features
- Fast response time, high throughput
- High availability and scalability
- Huge transaction rates, simple data
- Connectivity to Big Data Analytics and Discovery
Use Case – Online Display Advertising

• **Problem**
  – Very low latency requirements – Publishers require < 75 ms response time from the ad serving platform
  – Extreme data velocity – Multi-millions of requests per second
  – Highly available – 24/7 sites
  – Revenue maximization – Deliver the most relevant ad to maximize revenue

• **Solution – Where to use a NoSQL Database?**
  – Cookie store – NoSQL database used to store cookies and associated behavioral segments
  – Track behavioral data – Beacons utilized during browsing to store timestamp, frequency, and behavioral segments by cookie
  – Optimize ad delivery – Recency, frequency, and behavioral segments used to determine optimal ad to deliver to user
Use Case – Online Display Advertising

Architecture
Online Display Advertising

Using the right tool for the right job

- **Oracle NoSQL Database**
  - Low latency high volume
    - Millions of ad serving requests per minute or second
    - Stringent latency requirements from publishers
  - Loose consistency
    - Cookie data used for ad targeting – Increase probability that user will click on ad

- **Oracle Database**
  - Campaign booking information – hundreds of users
  - Real time business metrics for publishers and advertisers
  - Business financials for ad serving company
    - Year to date revenue, quarter over quarter etc., Billing, SOX reporting for public companies

- **Hadoop**
  - Unique visits (select count(distinct)) over many terabytes of data
  - Inventory forecasting across behavioral segments
Use Case – Rich Web 2.0 Applications

• Problem
  – Low latency requirements – Page loads in less than one second to retain users. Potentially hundreds of queries to populate page data.
  – Ease of scale – Automatic partitioning/scaling as user base grows
  – Highly available – 24/7 operations (e.g. Amazon.com)
  – Efficient state management – Application state info must extremely efficient

• Solution
  – Cookie store – A NoSQL database is utilized to store state and recommendation information (derived offline from Hadoop) by cookie
  – Profile management – rich user profile and history stored in NoSQL
  – Recommendations and initial state – Offline recommendations as well as initial state (e.g. recent products browsed) are read at page load time
  – State updated – The user’s state is updated in real time as browsing occurs
Rich Web 2.0 Applications - Overall Solution Architecture

Hadoop Cluster

App State Store

Oracle Database

Order Management & Financial Reporting

Nodes

http://www.example.com
Rich Web2.0 Applications

Using the right tool for the right job

- **Oracle NoSQL Database**
  - Low latency high volume
    - Millions of store browsing requests per minute
    - Predictable page rendering times - Less than 1 secs
  - Loose consistency
    - Cookie data used for UI customization

- **Oracle Database**
  - Order booking – Customer already “acquired”
  - Master product catalog
  - Business financial reporting
    - Year to date revenue, quarter over quarter etc., Quarterly auditing, SOX reporting for public companies

- **Hadoop**
  - Product recommendations
  - Probability of purchase
• Goal
  – Deliver a personal experience to every user
  – Each user profile must be retrieved and updated with minimal latency

• Challenge
  – Need to service this at web scale
  – 100k’s customers buying 100k’s movies

• Value
  – millisecond latency, on-demand scalability, high-throughput, lowest $/ops
Why Oracle NoSQL Database?

✓ Easy to use, easy to manage

✓ Scalable, Available, Predictable Latency

✓ A NoSQL Database from a vendor you trust
Oracle NoSQL DB Resources

- NoSQL DB Use Cases, White Papers, Data Sheets, Benchmarks

- NoSQL DB Documentation

- NoSQL DB Downloads

- NoSQL DB OTN Forum
  http://forums.oracle.com(forums/forum.jspa?forumID=1388)

- NoSQL DB version 2.0 Features

- OU Training Classes
Oracle NoSQL DB Resources

- Great Application Development article on OTN: http://www.oracle.com/technetwork/articles/bigdata/oracle-nosqldb-appdev-1891870.html
Questions & Discussion
APPENDIX
Big Data Acquisition Characteristics
Where should we put all that data?

<table>
<thead>
<tr>
<th>Batch-Oriented</th>
<th>Real-Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process data to use</td>
<td>Deliver a service</td>
</tr>
<tr>
<td>Bulk storage</td>
<td>Fast access to specific record</td>
</tr>
<tr>
<td>Write once, read all</td>
<td>Read, write, delete, update</td>
</tr>
<tr>
<td>Distributed, parallel bulk processing</td>
<td>“Last Mile” to the customer</td>
</tr>
</tbody>
</table>
How much throughput do you need?
NoSQL DB has throughput even for the largest players

API Billionaires Club

- twitter: 13 billion API calls / day (May 2011)
- Google: 5 billion API calls / day (April 2010)
- facebook: 5 billion API calls / day (October 2009)
- Netflix: 1.4 billion API calls / day (May 2012)
- ACCU WEATHER: 1.1 billion API calls / day (April 2011)
- KLOUT: 1 billion API calls / day (May 2012)
- eBay: 1 billion API calls / day (Q1 2012)
- Sabre: 1 billion API calls / day (January 2012)
Oracle NoSQL Database
Major-Minor Key Paradigm

/major/key/components/ - /minor/key/components

- /Smith/Bob/-/birthdate
- /Smith/Bob/-/phonenumber
- /Smith/Bob/-/image
- /Smith/Bob/-/userID

- /Smith/Richard/-/birthdate
- /Smith/Richard/-/phonenumber
- /Smith/Richard/-/image
- /Smith/Richard/-/userID

- /Wong/Bill/-/birthdate
- /Wong/Bill/-/phonenumber
- /Wong/Bill/-/image
- /Wong/Bill/-/userID

Oracle NoSQL Driver
Partitions and Shards

Partition# = Hash("Major Key")

Shard001
Partition001
K-V
K-V
K-V

Partition002
K-V
K-V
K-V

Partition100
K-V
K-V
K-V

Shard002
Partition101
K-V
K-V
K-V

Partition102
K-V
K-V
K-V

Partition200
K-V
K-V
K-V

Shard003
Partition201
K-V
K-V
K-V

Partition202
K-V
K-V
K-V

Partition300
K-V
K-V
K-V

Partition300
/home/john
Minor Key
Images
birthday
Value
001.jpg
002.jpg
1970-1-1
Simple Administration

- Web-based console and CLI commands
- Manages and Monitors
  - Configuration changes
  - Load: Number of operations, data size
  - Performance: Latency, throughput. Min, max, average, trailing, …
  - Events: Failover, recovery, load distribution
  - Alerts: Failure, poor performance, …
Elasticity
On-Demand Cluster Expansion

On Demand

- Increase Data Capacity
  - Add more storage nodes
  - New shards automatically created
- Increase Data Throughput
  - More shards = better write throughput
  - More replicas/shard = better read throughput
Rebalance an Unbalanced Store

Improve Performance

- Replication nodes move from over-utilized to under-utilized storage nodes
- Number of shards and replication factor remain unchanged
Oracle NoSQL DB Value Proposition

• Designed for the “last mile” problem
• Production-ready NoSQL database
• Built on industry-tested storage and replication technology (Berkeley DB)
• Integrates with Oracle and Open Source technology
• Supported by Oracle
Hardware and Software

Engineered to Work Together