



CouchBase @ eBay

John Kanagaraj, Global Data Infrastructure, eBay

October 6th, 2014

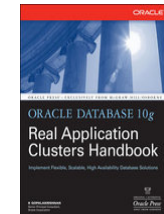
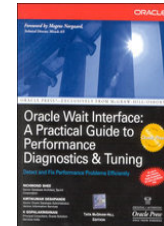
Agenda

- Introduction to **ebay**
- Why Couchbase?
- Use cases
- Current Couchbase deployments
- Challenges faced (and overcome!)
- Operationalizing Couchbase
- What's in the future
- Q & A



Speaker Qualifications

- Data Architect / Database Engineer at eBay Inc.
- Worked with Oracle Databases and UNIX for too long 😊
- Frequent speaker, Author and Technical editor
- Started working with NoSQL H2 2013; CouchBase early 2014
- Admin experience with CouchBase and other NoSQL products
- <http://www.linkedin.com/in/johnkanagaraj>
- See my “Oracle vs NoSQL” slide deck on SlideShare/LinkedIn



About eBay Inc.

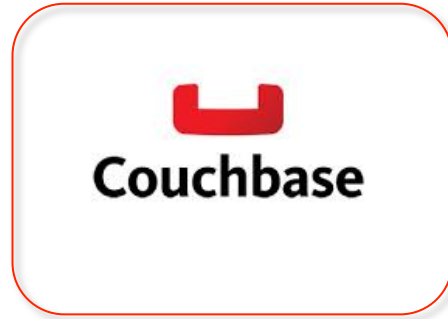


eBay Marketplaces Scale

- 100+ million active buyers and sellers
- 600+ million items
- 2+ billion page views a day
- 80+ billion database calls a day
- 5+ petabytes of site storage capacity
- 80+ petabytes of analytics storage capacity



Database landscape at eBay Inc.



- Hundreds of databases across the site using multiple technologies
- Choosing the right technology for the right use case is key to success



Why choose NoSQL (and Couchbase in particular!)

• Challenges of traditional RDBMS

- ACID overheads inhibit scalability
- Lack of native sharding and replication
- Need for “schema before write”
- Usually higher base cost

• Advantages of traditional RDBMS

- ACID is essential in many cases!
- Supports complex data models (e.g. ERP)
- Mature technology and ecosystems
- Skills widely available

• Advantages of NoSQL (Couchbase)

- Highly scalable at lower cost, quickly
- Web-scale K-V read/write performance
- Flexible schema
- Both open source & Enterprise models

• Challenges of NoSQL

- Lack of ACID and Transactions
- CAP is real in distributed databases
- Evolving and maturing but fragmented
- Skill sets not widely available (yet)



Why Couchbase?

- Both Open source and Enterprise models available
- Memcached API – Ideal for Caching use cases
- Low latency (< 1ms) on write and read
- Horizontally scalable model
 - 300,000+ Reads / sec on 1 server
 - 25,000 Writes / sec on 1 server with low replication queue
 - Capable of 100,000 Writes / sec burst writes on 1 server
- Ejection of values for larger and/or inactive data sets (changes in 3.0!)
- Easily and dynamically add capacity without down time
- Client support for multiple languages
- Cross data center replication with multi-master clusters
- Technology still maturing but currently best fit for caching

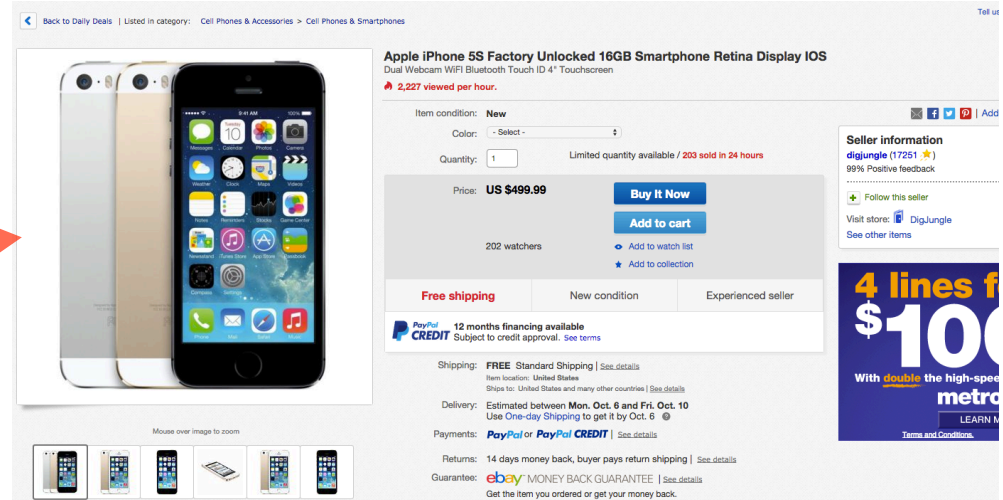
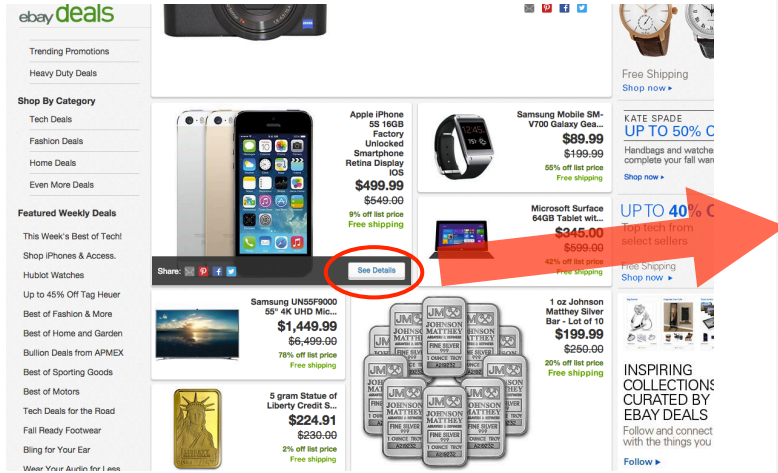


Object Caching Patterns

- Read Buffering (Speeds/optimizes access to System-of-Record – SoR)
 - Frequently accessed data: L2 Distributed/Out of process cache
 - Infrequently changing data: Reference Data cache
 - Result set caching: Cache expensive computations
- Write Buffering (Writes directly only to cache)
 - State preservation: Session cache/token store; cookies;
 - Event store for buffering expensive writes:
 - Stage “bursty” logs
 - In-flight CEP (Complex Event Processing)
 - Rate limiters



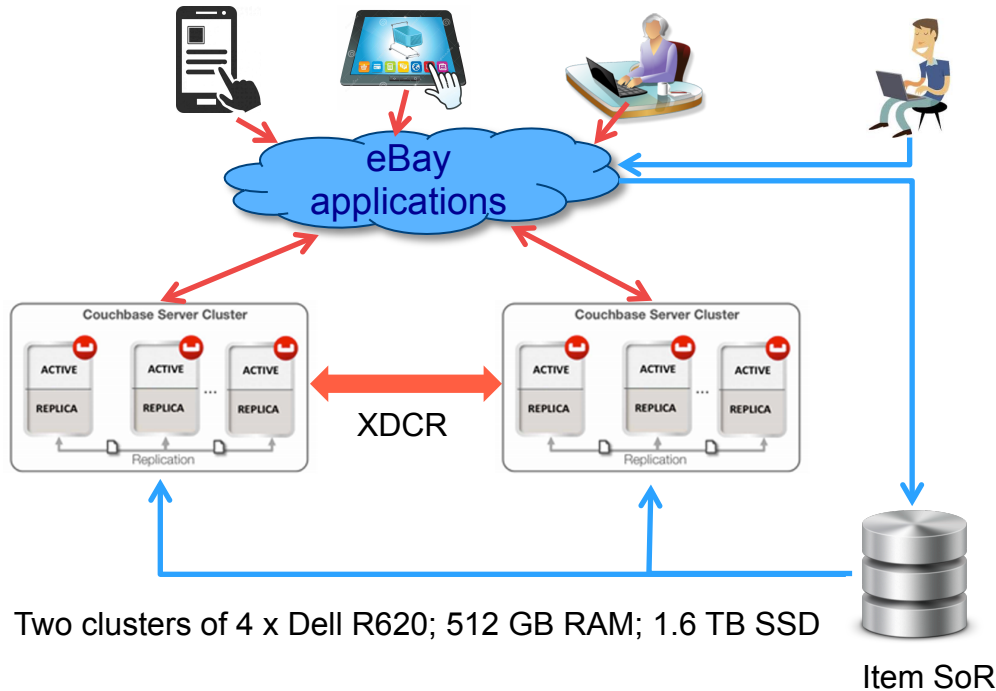
A Tale of Two Deployments: “View Item” Read Buffering



- View Item Page - Listing Cache
 - Speeds up detailed page view (upon click on item)
 - Enhances user experience



1. Listing Cache Couchbase deployment



- Items listed by seller
- Inserts/Updates SoR
- Couchbase based clusters caches some item details and item metadata
- Multichannel, multiflows supported
- Reads sped up when View Item Details is clicked
- DB Event processor updates Item changes to cache + cache miss updates cache
- Bi-directional XDCR between Datacenters
- Item expires in SoR and cache after fixed interval

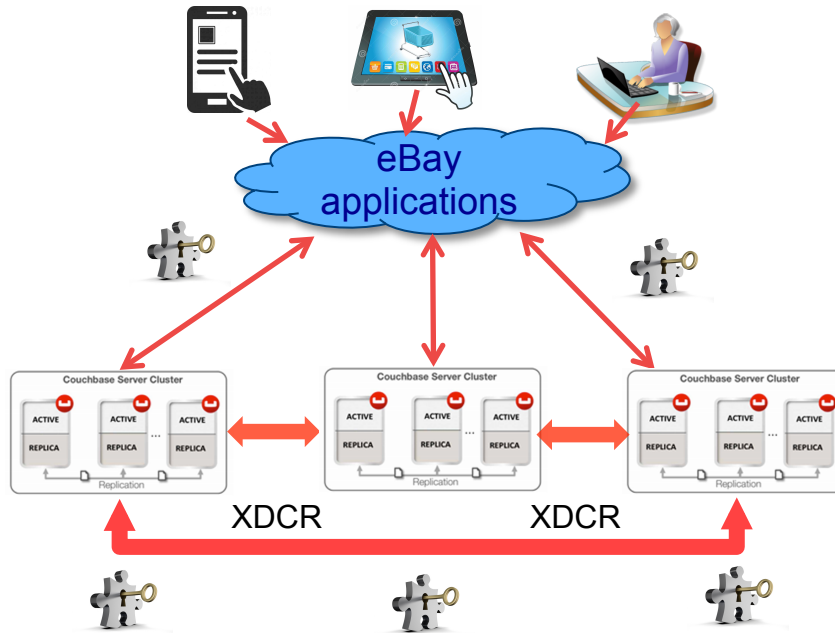


A Tale of Two Deployments: Session/Token Store

- Auth Token store:
 - Auth Token created during login
 - Written to Couchbase and read back during re-authorizations
 - Sets up new platform for creating and extending new auth policies
 - Created as a service layer to support both legacy and new flows



2. Session/Token Store Couchbase deployment



Three clusters of 6 x Dell R620; 128 GB RAM; 1 TB HDD

- User logs in to eBay
- Token generated
- Tokens written to Couchbase based clusters
- Tokens reread for session validation
- Bi-directional XDCR between three datacenters
- Tokens expire after fixed interval
- Multichannel, multiflows supported



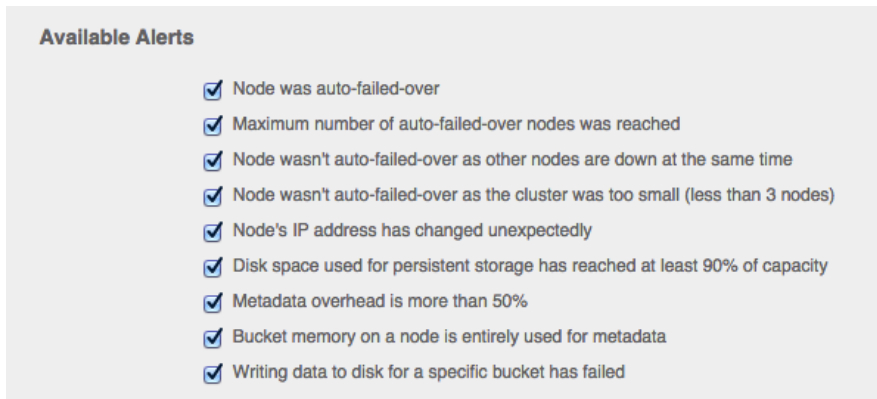
Challenges faced (and overcome!)

- Earliest of Couchbase use cases: Needed quick skills ramp up (Dev/DBA)
- Couchbase Support and Engineering provided excellent support throughout
- Nodes going into pending state intermittently
 - Tracked down to a new XDCR bug in 2.5.1
 - Hotfix delivered recently: Testing/Implementation in progress
- Reconnect from Clients fails intermittently
 - Tracked down to a new bug in 1.4.3 and below with Carrier Publication Protocol
 - Client version 1.4.4 delivered recently: Testing/Implementation in progress
- Inconsistency in XDCR replication
 - Analysis in progress: Testing with 2.5.1 XDCR Hot fix



Operationalizing Couchbase – Points to ponder

- Provisioning
 - Internal or External?
 - If internal, VM or Bare Metal?
- Monitoring, Alerting and Metrics
 - Use Alerts from CB Cluster console
 - Integrating to your Ops board
 - Collecting/Storing/Alerting using cbstats and OS metrics (Nagios, Zabbix)
- Performance tracking
 - Measure both client side and server side
 - Need configurable logging on client side



Operationalizing Couchbase – Points to ponder

- Capacity management:
 - Who pushes the envelope? RAM, CPU, Disk, I/O, Network
 - Capacitize for failure modes: # of nodes, # of copies, %age load shift on failure
 - Trend metrics and review periodically
 - Cater for both organic growth and peaks
- XDCR and Data protection
 - XDCR based failovers/protection: Bi-directional/one way? Cross-colo? N/W bandwidth?
 - cbackup vs. file copy (needs identical cluster/vbucket config with offline restoration)
- Application Behavior at Scale
 - Connection management: # of clients; connection storm during startup
 - Failure modes: single node vs. whole cluster
 - Fail/Open vs. Fail/Close
 - Timeouts and the ripple effect



Operationalizing Couchbase – Points to ponder

- Data Modeling and Governance
 - Designing and Evolving your data model: Access patterns, Indexing, Bucketing
 - Managing type and sensitivity of data being stored
- Security
 - Password and access management
 - Layered security
- Data Lifecycle management: Implement retention policy – use TTLs
- Change Management: Process for changing configuration and settings
- Support: Internal skills and External Technical Support from Couchbase
- Many more: Licensing, Cost management, ETL, etc.



What's coming

- Number of use cases lining up for Couchbase
 - eBay Global Preferences cache
 - eBay Cookie store
 - eBay Mobile Identity cache
 - Taste Graph
 - Notification Inbox
- Ramp up and Holiday preparedness
- Internal awareness and training campaigns
- Couchbase 3.0!!!!





Q & A