ORM, No-SQL & Vietnam

Sean Corfield
World Singles llc
Agenda

• What is ORM?
• What is "No-SQL"?
• Why Vietnam?
• ORM & "No-SQL" & CFML
You Might Prefer...

• Apache Ninja Crash Course
• HTML5 Jedi Powerness with ColdFusion 10
• Navigating the Front End Ocean
• Making ColdFusion Apps Highly Available
  • Deep Dive - two hours starting now!
Me

- Functional Programming in the 80's
- Object-Oriented Programming in the 90's
- Web / Dynamic Programming in the 00's
- Mostly Clojure today
Me & ORM & NoSQL

• Built production apps with Reactor, Transfer, Hibernate and others
• Written several data mappers / mini-ORMs
• Using MongoDB in production (2 years)
• Old enough to remember Versant, Poet, etc
Agenda

• What is ORM?
• What is "No-SQL"?
• Why Vietnam?
• ORM & "No-SQL" & CFML
What is ORM?

- Object-Relational Mapping
- We program with Objects
- We store data in a Relational System
Object Systems

• Identity
• State - Equivalence
• Behavior
• Encapsulation
Relational Systems

- Attribute - Column (Name + Type)
- Tuple - Row
- Relation - Attributes + Tuples
- Relation Value - Relation + Tuples
- Relation Variable - Named Relation
Mapping

- Identity?
- State <=> Tuple
- Behavior - in-memory only
- Encapsulation - in-memory only
Identity

- Object Identity != Relational Identity
- State augmented with PK
- Objects now have two identities :(
CRUD

• One class per table
• Design class => Create table
• Create object & save => Insert row
• Modify object & save => Update row
• Delete object => Delete row
CRUD

• Select row => Create object & load
  • Must select a single row
CRUD

- Select multiple rows?
- Query (result set) - not OO
- Array of objects - inefficient
- Iterating Business Object - workaround
Relationships

• Objects related by identity
  • Parents know about their children
  • Has-a - parent has child reference
  • Has-many - parent has array of children
• Unidirectional
Relationships

• Relational model uses keys
  • Children know of parents
  • Foreign key relation
  • One-to-many / Many-to-one
  • Non-directional - join is "external"
Friction

- The mapping is incomplete / artificial
- Object inheritance complicates things
  - Table per class?
  - Table per concrete class?
  - Table per class family?
Friction

- Schema design
- Ownership?
  - Code is fluid for developers
  - Database is not so fluid for DBAs
  - DBAs concerned with performance
Friction

• Schema design
  • Metadata is duplicated
    • Column names, types, constraints etc
    • Class attribute names, types, structures
  • No constraint enforcement
Hibernate as ORM

- Lots of annotations in code
- Might need tweaking in XML
- "session" management
- Transaction management
- Caching implications
- Cascading operations
ACID & Transactions

• Atomic, Consistent, Isolated, Durable
• Transactions - commit / rollback
• Managed / enforced by database only
• Objects have NONE OF THESE!
  • Software Transactional Memory?
Hibernate as ORM

- Reporting and complex queries
  - Use HQL?
  - Revert to raw SQL?
- Don't want objects / encapsulation?
ORM Query Language

- Query by example
  - Easy to use / understand
  - Too simple for many situations
ORM Query Language

- Query by API
  - Build queries as objects
  - Complex API, Builder pattern
- To verbose for many situations
ORM Query Language

- Query by language
  - Duplicates SQL (as HQL)
    - Several subtle differences
  - It's still just a mapping
    - So it's likely to be incomplete
Agenda

• What is ORM?
• **What is "No-SQL"?**
• Why Vietnam?
• ORM & "No-SQL" & CFML
What is "No-SQL"?

- We'll get to that in a minute...
Agenda

• What is ORM?
• What is "No-SQL"?
• Why Vietnam?
• ORM & "No-SQL" & CFML
Let's visit Vietnam

- "Object/relational mapping is the Vietnam of Computer Science" -- Ted Neward, 2004
- He wrote this up in more detail in 2006:
Let's revisit Vietnam

• Jeff Atwood posted a followup focusing on Ted's summary points:

• http://www.codinghorror.com/blog/2006/06/object-relational-mapping-is-the-vietnam-of-computer-science.html

• "I think the only workable solution ... is to ... abandon relational databases, or abandon objects."
But we won the Vietnam war, right?

- Why is ORM like the Vietnam war?
- We think it's a solvable problem...
- With just a little more work...
- Despite past problems, we keep going...
- Please read Ted Neward's summary at least!
Agenda

• What is ORM?
• **What is "No-SQL"?**
• Why Vietnam?
• ORM & "No-SQL" & CFML
What is No-SQL?

- Alternative data storage approaches
- Ad hoc / API queries rather than SQL
- Optimized for specific use cases
- Usually highly scalable
  - Easy to cluster
  - Easy replication
What is No-SQL?

- Four primary types of data store
  - Key-Value-based
  - Column-based
  - Graph-based
  - Document-based
What is No-SQL?

- Key-Value-based
  - Redis
  - MemcacheDB
  - Riak (adds map/reduce)
What is No-SQL?

• Column-based
  • BigTable (Google)
  • HBase (part of Apache Hadoop)
• Cassandra (Facebook)
What is No-SQL?

- Graph-based
  - Neo4j

- Optimized for edge retrieval (like joins)

- Supports ACID (surprisingly)
What is No-SQL?

- Document-based
  - MongoDB
  - CouchDB (now CouchBase)
- Closest to SQL / Relational model
Document-based

- Closest match to objects
  - One document per object
- Simple CRUD "just works"
- Schema is flexible / heterogeneous
  - It is not "schema-less"
Document-based

- Relationships
  - Can be embedded
    - Document (one-to-one)
    - Array (one-to-many)
- Stored separately
  - Managed by Object ID in app code
Document-based

• No transactions
  • Not actually needed as much as you think
  • Gregor Hohpe - "Your Coffee Shop Doesn't Use Two-Phase Commit"
  • http://www.eaipatterns.com/docs/IEEE_Software_Design_2PC.pdf
Document-based

- No joins
- Embedded documents
- Aggregation queries
Document-based

• Schema design can still be critical!
• Design to match queries, not data
• Pre-aggregate computed values
  • Atomic increment updates
• Reports are simple queries
Agenda

• What is ORM?
• What is "No-SQL"?
• Why Vietnam?
• ORM & "No-SQL" & CFML
ORM & CFML

• ColdFusion 10 / Railo 4 integrated Hibernate to provide persistence for a CFC-Relational-Mapping

• Simple CRUD is very slick

• Complex models expose more and more of Hibernate for you to learn
"No-SQL" & CFML

- Mostly you can just use the Java libraries
- For MongoDB, there's CFMongoDB
  - http://cfmongodb.riaforge.org
Let's see some code!
Summary

- Mapping Objects to/from Relational DBs has a number of inherent flaws which become more painful as your system grows in complexity.

- Document-based "NoSQL" data stores relieve many of the mapping problems but have trade-offs around complex queries.

- The relief may be worth the trade-offs...
Q & A?

- @seancorfield
- http://corfield.org
- sean@corfield.org