Polyglot Persistence in Today’s Data World

Kimberly Wilkins
Principal Engineer – Databases
ObjectRocket by Rackspace
www.linkedin.com/in/wilkinskimberly,
kimberly.wilkins@rackspace.com,
@dba_denizen
Background

- 18+ years working on various database platforms
- Mainly Oracle (databases, RAC, Enterprise Manager, GoldenGate Replication, DataGuard, Database Vault, Exadata)
- MongoDB NoSQL and Big Data techs at OR
- Industries – early online auctions, gaming, social media
- Specialties – re-architect enterprise database environments, infrastructure implementations, RAC, replication, system kernels, database storage
- Re-engineered the database infrastructure for SWTOR – Star Wars The Old Republic MMO game
- Supporting customers large and small and presenting for OR/Rackspace
Overview

- What is Polyglot Persistence
- Why Do We Need It
- Data Technologies Available
- IoT Impacts – Data Store Techs
- 3 V’s and CAP
- Techs Described
- Questions To Ask When Designing
- Do’s, Don’t’s, More
- Connectors
- Polyglot Morphing
What is Polyglot Persistence

• Based off of premise of Polyglot Programming
• Write Applications in a mix of languages
• Take advantage of individual strengths
• Avoid their weaknesses

The Languages of the 4th Millennium

Python  26%
JavaScript  23%
C#  20%
Java  15%
C++  13%
What is Polyglot Persistence...

• Different Database techs for different use cases and applications
• Right DB technology for the task – use the right DBMS tool for the job
• NoSQL Mantra ...
• Benefits must outweigh costs
Questions To Ask

• Does it work for specific application needs?
• Does it work for individual needs of your overall business?
• How well for all of your applications?
• How much complexity can you support?
• What will added cost be – in hardware/software/personnel
Explosive Growth of Big Data

Gartner - Global information volumes rising over 59% each year
IDC - Big Data market "expected to grow from $3.2 billion in 2010 to $16.9B in 2015."

For YOU – career options, career changes
Facebook 500 TB daily
Internet 1 Exabyte created per day
Brontobytes, Zetabytes -IoT

www.objectrocket.com
New Industries, New Sources, Much More Data
Internet of Things/Everything

• Last 2 years - more data than all of human existence in the years prior
• Multiply \(^\wedge\wedge\) than we can imagine
• IoT growth is driving adoption
• Up to 20 billion devices – connect, log, automate
• Cell Phones, Smart Phones, Mobile Phones
• \(\Rightarrow\) 5 billion people calling, texting, tweeting, and browsing worldwide
Using Multiple Data Store Technologies

Individual RDBMS and NoSQL Database Technologies by themselves cannot handle today’s diverse application and data needs.
Mixed Use and Source Example
The 3 V’s: Velocity, Variety, Volume
Cap Theorem

Data Models
- Relational (Comparison)
- Key-value
- Column-oriented / Tabular
- Document oriented

Availability
- Each client can always read and write

Consistency
- All clients always have the same view of the data

Partition Tolerance
- The system works well despite physical network partitions

Pick 2

CA
- RDBMSs (MySQL, Postgres, etc)
- Aster Data
- Greenplum
- Vertica

AP
- Dynamo
- Voldemort
- Tokyo Cabinet
- KAI
- Cassandra
- SimpleDB
- CouchDB
- Riak

CP
- BigTable
- Hypertable
- HBase
- MongoDB
- Terrastore
- Scalaris
- Berkeley DB
- MemcacheDB
- Redis
Possible Right Tools – Data Structure

- Relational Databases – Oracle, MySQL, Postgres
- Key Value Stores – Redis, Riak
- Column Stores – Cassandra, Hbase
- Document Stores – MongoDB, CouchDB
- Graph databases – Neo4J
- Search Engines – Elastic, Solr
Relational Databases

• Use when:
  - Your dataset is relational
  - Strong consistency needed - ACID
  - Access patterns are unknown

• But ... Doesn’t scale well horizontally

• Use Cases:
  - Due to early adoption and heavy legacy usage - everywhere
  - Payment systems
Key/Value stores

• Use when:
  • - Operations are based on the key
  • - Data is not highly related
  • - Basic CRUD needed

• But... Complex queries are painful

• Use cases:
  - Session Data
  - User Profile/Preferences
  - Shopping carts
Key/Value stores – cont’d

• A big hash map associative array
  - Very simple, One key <-> One value
  - Very fast read/write
  - No secondary indexes

• "Key": (VRN) => {value (car facts)}

• [make#Ford
• {YYY0000} => model#Fiesta
• year#2010]
Document Store Databases

- Nested structures of keys and their values
  - Very flexible schema (JSON, XML)
  - One key One value but value is visible to queries
  - Supports hierarchical data
  - Supports secondary indexes

- {“id”: (VRN)} => {“document” (car facts)}

  - { “make”: Ford”,
  - {YYY0000} => “model”:”Fiesta”,
  - “year”:2010 }
Document Store Databases

• Use when:
  - You don’t know much for the schema – Unstructured, Semi-structured
  - Unstructured and Heterogeneous data

• But ...
• Joins and references are tricky
• De-normalization requires more space

• Use cases:
• - Product Catalog
• - CMS - Adobe, etc.
• - Event logging from different sources
**Columnar Store Databases**

- Use when:
  - Big Data (Huge write volumes)
  - Versioning (Time-series data)

- But...
  - Know your statements in advance
  - Schema design is not trivial

- Use cases:
  - Time series data
  - Bidding platforms
  - Playlists
Document vs Column Store Databases

MongoDB (document store) better with:
- When ‘schema’ will be fluid, changing
- With heavy read loads or mixed loads

Cassandra (column store) better with:
- When ‘schema’ will be fixed, NOT changing, set
- With heavy write loads – faster, append only
Graph Databases – 6 Degrees of Separation..

• Use when:
  - Highly interconnected data
  - Define explicit relationships and need traversal queries
• But ... Doesn’t scale well horizontally

• Use Cases:
  - Where 3rd degree (or higher) relationship needed
  - Social Media - Queries like friend of a friend, likes and dislikes
Questions to Ask when Picking a Datastore Technology

- Is Data Structured or Unstructured
- How is the Data Connected - likes
- Size of the data – now? Later?
- Read/Write Ratio – heavy read/write/mixed?
- Is Access Pattern Consistent All of the time – Monthly reporting, heavy queries in burst, insert batches
Questions to Ask when Picking a Datastore Technology

• Do I Need full DR?
• Security/Encryption
• Backups – immediate, delayed
• Immediately available or delayed

• Language Support
• Driver Availability
• Community Support
• Connector Availability
• Monitoring – plug-ins
• Driver Availability
Challenges: Beware Polyglot Persistence

• Not All Roses
• Added Complexity
• More systems to manage
• More expertise required

• Interactions
• Network Connectivity
• Timing Issues
• Where to store which data for which apps/data stores
• Additional Development Efforts, Upgrades, Testing

www.objectrocket.com
Challenges: Plan – Architect Ahead

- Decide which to use for what
- Possible Migration Headaches if you chose wrong on any level 1st
- More systems to deploy
- More hardware to provision
- More configs to standardize
- More Connections to manage
- More troubleshooting – different techs
- More personnel to support techs
- More training for support
Do’s : Plan Ahead

- KISS
- Keep system streamlined
- Keep system as lean as possible
- Test and benchmark as much as possible
- Stick to more proven use cases
- Push boundaries or your customers will do so for you... ;-)

www.objectrocket.com
Connectors...

ObjectRocket currently has the Elastic to MongoDB Connector available:

http://objectrocket.com/data-connectors/

Other Community connectors are also available.
Polyglot Morphing...

What we have Today, will almost surely change tomorrow.....
Current Pluggable Engines
Questions?
Thank you!

**Address:**
401 Congress Ave  Suite 1950
Austin, TX 78701

**Support:**
1-800-961-4454

**Sales:**
1-888-440-3242

www.objectrocket.com