Post-relational databases

What's wrong with web development?

Dobrica Pavlinušić

http://blog.rot13.org

FOI, Razmjena Vještina, Varaždin, 2010-12-10
Who am I?

- Web programming since 1995 using FLOSS
- Languages: ρhρ, perl, JavaScript
- Databases: PostgreSQL, MySQL, CouchDB
- 5+ years of experience as system architect on large intranet content portal
- http://blog.rot13.org

- Big question: are we solving same problems over and over again?
- Model-View-Controller pattern
Back in mainframe days...

- All logic in COBOL application
- Similar to Clipper in DOS
- Payroll-type applications
GUI Client/Server RDBMS

- **tabular** normalized data (3NF)
- SQL to query and modify data (static schema)
- logic in controller and RDBMS (validation)
"network is computer" – browser thin client
Logic in controller, RDBMS & view (JavaScript)
Trees (XML)? Self-referencing data?
AJAX JSON REST Web2.0

- JSON without schema (or verification!)
- Logic in JavaScript on client and server
- Turtles all the way down reduces complexity
Perfect web stack

- HTTP REST API with JSON
  - GET, PUT/POST, DELETE
- JavaScript queries using map/reduce
  - Real-world dirty data from views
- Serve application from HTTP database (content-type support)
- Two-way data binding forms <-> JSON
  - HTML is only template we need!
- Single language: JavaScript (+jquery)
- "simple app in single afternoon" benchmark
CouchDB

- Designed for replication over HTTP
  - applications replicated with data (off-line)
- Custom views, requests and _changes feed
What if web browsers where written for web applications today?
- html compiler inside browser
- ng:repeat and friends to template objects
- Objects persisted using REST to server

http://docs.angularjs.org/

Demo
- printer overview
- conference submission app
Mojolicious – web in the box (perl)
  - http://mojolicious.org/
Mojo::Client – chunked HTTP client
  - Perfrect for CouchDB integration
https://github.com/dpavlin/angular-mojolicious
REST API for Angular's $resource
  - CouchDB proxy or static JSON files
Replication with getangular.com service
  - not API compliant, uses newer version
Helpers to quickly prototype with angular
CouchDB triggers

- Finite-state machine inside document
- Multiple workers - FSM for lock and status
- filter, trigger, commit on _changes feed
- Perfect for async tasks
  - executing shell
  - sending e-mail
CouchDB full-text search

- Implemented as filter-only trigger
- KinoSearch – full-text search, base for Apache lucy (Lucene in C)
- Trigger which delete/add documents
  - Unroll structures into.flat.schema
- CouchDB external server to perform queries
CouchDB related projects

- BigCouch
  - https://github.com/cloudant/bigcouch
  - Consistant hashing, sharding
- ElasticSearch
  - http://www.elasticsearch.com/
  - CouchDB river
- GeoCouch
  - https://github.com/vmx/couchdb
  - Spartial index
Riak (search)

- **cluster** of machines!
- Amazon's dynamo model
  - r,w – eventual consistency
- Multiple map/reduce phases in single query
  - Ad-hoc, not cached, in parallel over cluster
- post-commit hooks (search)
- Links (REST traversal)
- HTTP and ProcolBuffers interfaces
- [http://www.basho.com/riaksearch.html](http://www.basho.com/riaksearch.html)
Overview

- Avoid complexity
  - Model: JSON
  - View: HTML
  - Controller: angular
- Solve offline problem: CouchDB
  - Attach async processes in the cloud!
- Attack interesting problems
  - Tree-data, (social) networks