Building Event Based Systems for the Real-Time Web

Paul Dix
http://pauldix.net
@pauldix
market.io

cofounder and CTO
The Talk
Event Based:
data update => do something
Batch:
need data => calculate
Real-Time:
data update => notification
Scheduled:
occasionally => do stuff
Event Based == Real Time
Monolithic

American Backroom (http://www.flickr.com/photos/41922098@N03/4247207167/)
Loosely Coupled
Scale

'Playingwithbrushes' (http://www.flickr.com/photos/82518118@N00/2280744328/)
Scale with Complexity

PhOtOnQuAnTiQuE (http://www.flickr.com/photos/67968452@N00/1876685709/)
Scale with Team Size

alexkess (http://www.flickr.com/photos/34838158@N00/3370167184/)
Definition by Example
SQL Databases
Event Based:
triggers
Batch:
count, sum, min, max
Real-Time?
counter cache
Scheduled?
cron, at, schtasks
Broken

nickwheeleroz (http://www.flickr.com/photos/nickwheeleroz/2474196275/in/photostream/)
Self Contained
Monolithic
Bad!

Road Fun (http://www.flickr.com/photos/21849473@N06/2550339131/)
Doesn’t Scale
Distributed Systems

cygnoir (http://www.flickr.com/photos/35034356212@N01/163671482/)
Event Based: messaging, publish subscribe, amqp
Batch:
MapReduce, Hadoop
Real-Time: messaging, publish/subscribe
Scheduled: delayed replication, data backups
The Web

cygnoir (http://www.flickr.com/photos/35034356212@N01/163671482/)
Event Based:
web hooks,
pubsubhubub, rsscloud
Batch:
polling
Real-Time:
web hooks, web sockets, pubsubhubbub, rsscloud
Scheduled:
check every 1 hour
Distributed (internal) Systems
Service Calls
Feed Reader (on fetch)
Feed Reader (on fetch)

- Update users reading list
Feed Reader (on fetch)

- Update users reading list
- Language Identification
Feed Reader (on fetch)

- Update users reading list
- Language Identification
- Named Entity Extraction
Feed Reader (on fetch)

- Update users reading list
- Language Identification
- Named Entity Extraction
- Cluster with other Articles
Feed Reader (on fetch)

- Update users reading list
- Language Identification
- Named Entity Extraction
- Cluster with other Articles
- Identify Trending Entities
Tightly Coupled

Steve aka Crispin Swan (http://www.flickr.com/photos/26811962@N05/37045553536/)
Publish/Subscribe (pubsub)
Exchanges
Queues
Routing Key
Exchange Type
Topic Exchange
<token>.<token>...
# - 0 or more wildcard
* - 1 wildcard
routing key:
feed_entry.insert

'notify'
Topic Exchange

Reading List
Queue
routing key: feed_entry.insert

'notify' Topic Exchange

Reading List Queue

Language Queue
routing key: feed_entry.insert

'notify'
Topic Exchange

- Reading List Queue
- Language Queue
- Named Entity Queue
- Clustering Queue
Include Data in the Message
Data is the API
Another Example:
error logging
routing key:
domU-12-31-39-07.feed_fetcher
binding:
*.feed_fetcher
binding:
#
#
Loosely Coupled
Event-Based
Real-Time
Web Hooks
GitHub
Post-Receive Hooks
**Service Hooks**

After setting up any of the following services, we'll send your commit information to them when you push to GitHub.

**Hooks**
- Post-Receive URLs (0)
- Basecamp
- CIA
- Campfire
- Email
- FogBugz
- FriendFeed

**Post-Receive URLs**

We'll hit these URLs with POST requests when you push to us, passing along information about the push. More information can be found in the Post-Receive Guide.
git push triggers hooks
POST to Post-Recieve URLs
{"before": "5aef35982fb2d34e9d9d4502f6ede1072793222d",
"repository": {
  "url": "http://github.com/defunkt/github",
  "name": "github",
  "description": "You're lookin' at it.",
  "watchers": 5,
  "forks": 2,
  "private": 1,
  "owner": {
    "email": "chris@ozmm.org",
    "name": "defunkt"
  }
},
"commits": [ {
  "id": "41a212ee83ca127e3c8cf465891ab7216a705f59",
  "url": "http://github.com/defunkt/github/commit/41a212ee83ca127e3c8cf465891ab7216a705f59",
  "author": {
    "email": "chris@ozmm.org",
    "name": "Chris Wanstrath"
  },
  "message": "okay i give in",
  "timestamp": "2008-02-15T14:57:17-08:00",
},
{ "id": "de8251ff97ee194a289832576287d6f8ad74e3d0",
  "url": "http://github.com/defunkt/github/commit/de8251ff97ee194a289832576287d6f8ad74e3d0",
  "author": {
    "email": "chris@ozmm.org",
    "name": "Chris Wanstrath"
  },
  "message": "update pricing a tad",
  "timestamp": "2008-02-15T14:36:34-08:00"}]
"after": "de8251ff97ee194a289832576287d6f8ad74e3d0",
"ref": "refs/heads/master"}
Data is the API
Continuous Integration
Bug Tracking
Campfire
Continuous Deployment
PubSubHubbub
RSS or Atom Feed

<link rel="hub" href="http://pubsubhubbub.appspot.com/"/>
Subscriber tells Hub
Hub verifies with Subscriber
Publisher notifies Hub
Hub gets from Publisher
Hub sends to Subscribers
Data is the API
OAuth + Web Hooks
Concerns
Don't make web-hook callbacks while a user is waiting.

Responsive
Retries

what if the subscriber doesn't respond?
retry logic
notifications
Security

use OAuth, SSL for sensitive data
Possibilities
Real-Time
Event Based
Are these really real-time?
Data Consistency
Eric Brewer’s CAP theorem

In Brewer’s CAP theorem, he talked about the relationship between three requirements when building distributed systems: consistency, availability, and partition tolerance.
consistency means that an operation either works completely or fails. This is also referred to as atomic.

Bug tracker example.
availability is pretty self explanatory. A service is available to serve requests.

Twitter example
when you replicate data across multiple systems, you create the possibility of forming a partition. this happens when one or more systems lose connectivity to other systems. partition tolerance is defined formally as "no set of failures less than total network failure is allowed to cause the system to respond incorrectly"
pick two
Werner Vogels’
eventual consistency

"Can have all three is a special form of weak consistency. If no new updates are made to an object, eventually all accesses will return the last updated value."
Design Around Eventual Consistency
In Closing...
Event-Based == Real Time
Pubsub
Loosely Coupled
Expose Data
Third Party Programmers
Extend and Improve
many heads are better...

shoothead (http://www.flickr.com/photos/66621443@N00/519240547/)
Questions?

Paul Dix
http://pauldix.net
@pauldix