Building a Highly Scalable, Open Source Twitter Clone

Dan Diephouse (dan@netzooid.com)
Paul Brown (prb@mult.ifario.us)

Motivation

- Wide (and growing) variety of non-relational databases.

  (viz. NoSQL — http://bit.ly/pLhqQ, http://bit.ly/17MmTk)

- Twitter application model presents interesting challenges of scope and scale.

  (viz. “Fixing Twitter” http://bit.ly/2VmZdz)

Storage Metaphors

- Key/Value Store

  Opaque values; fast and simple.

- Examples:
  - Cassandra* — http://bit.ly/EdUEt
  - Dynomite — http://bit.ly/12AYmf
  - Redis — http://bit.ly/LBtCh
  - Tokyo Tyrant — http://bit.ly/oU4uV
  - Voldemort - http://bit.ly/oU4uV

Key/Value

Key	Value
1	☁
2	☁
3	☁

Storage Metaphors

- Document-Oriented

  Unstructured content; rich queries.
- Examples:
  - CouchDB — http://bit.ly/JAgUM
  - MongoDB — http://bit.ly/HDDOV
  - SOLR — http://bit.ly/q4gyi
  - XML databases...

Document-Oriented

```
ID="dan-tweet-1",
TEXT="hello world"
```

```
ID=dan-tweet-2,
TEXT="Twirp!",
IN-REPLY-TO="paul-tweet-5"
```

Storage Metaphors

- Column-Oriented

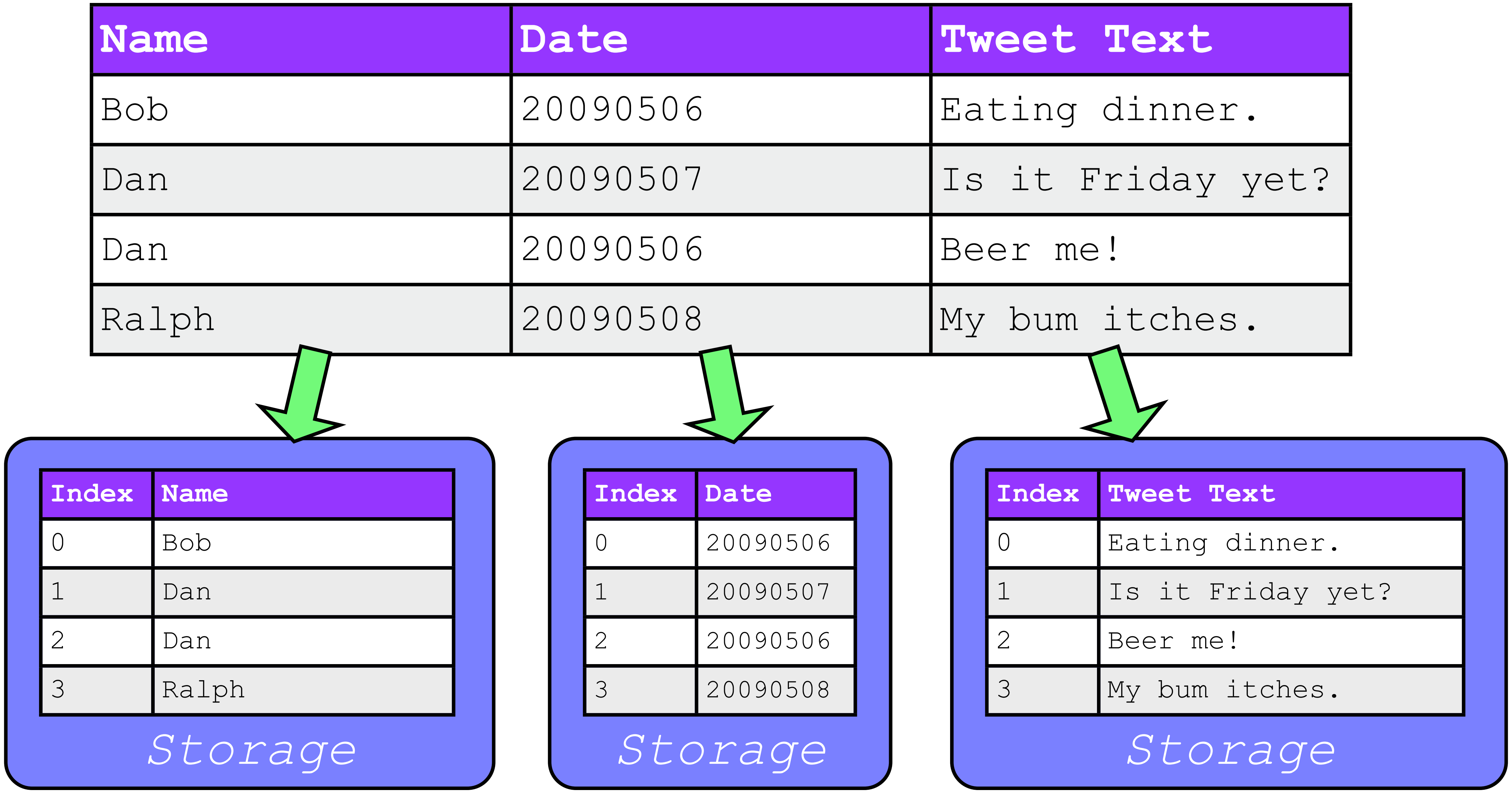
  Organized in columns; easily scanned.

- Examples:
  - Cassandra* – http://bit.ly/EdUEt
  - BigTable – http://bit.ly/QqMYA

    (available within AppEngine)
  - HBase – http://bit.ly/Zck7F
  - SimpleDB – http://bit.ly/toh0P

    (Typica library for Java – http://bit.ly/22kxZ4)

Column-Oriented



Name	Date	Tweet Text
Bob	20090506	Eating dinner.
Dan	20090507	Is it Friday yet?
Dan	20090506	Beer me!
Ralph	20090508	My bum itches.

Index	Name
0	Bob
1	Dan
2	Dan
3	Ralph

Index	Date
0	20090506
1	20090507
2	20090506
3	20090508

Index	Tweet Text
0	Eating dinner.
1	Is it Friday yet?
2	Beer me!
3	My bum itches.

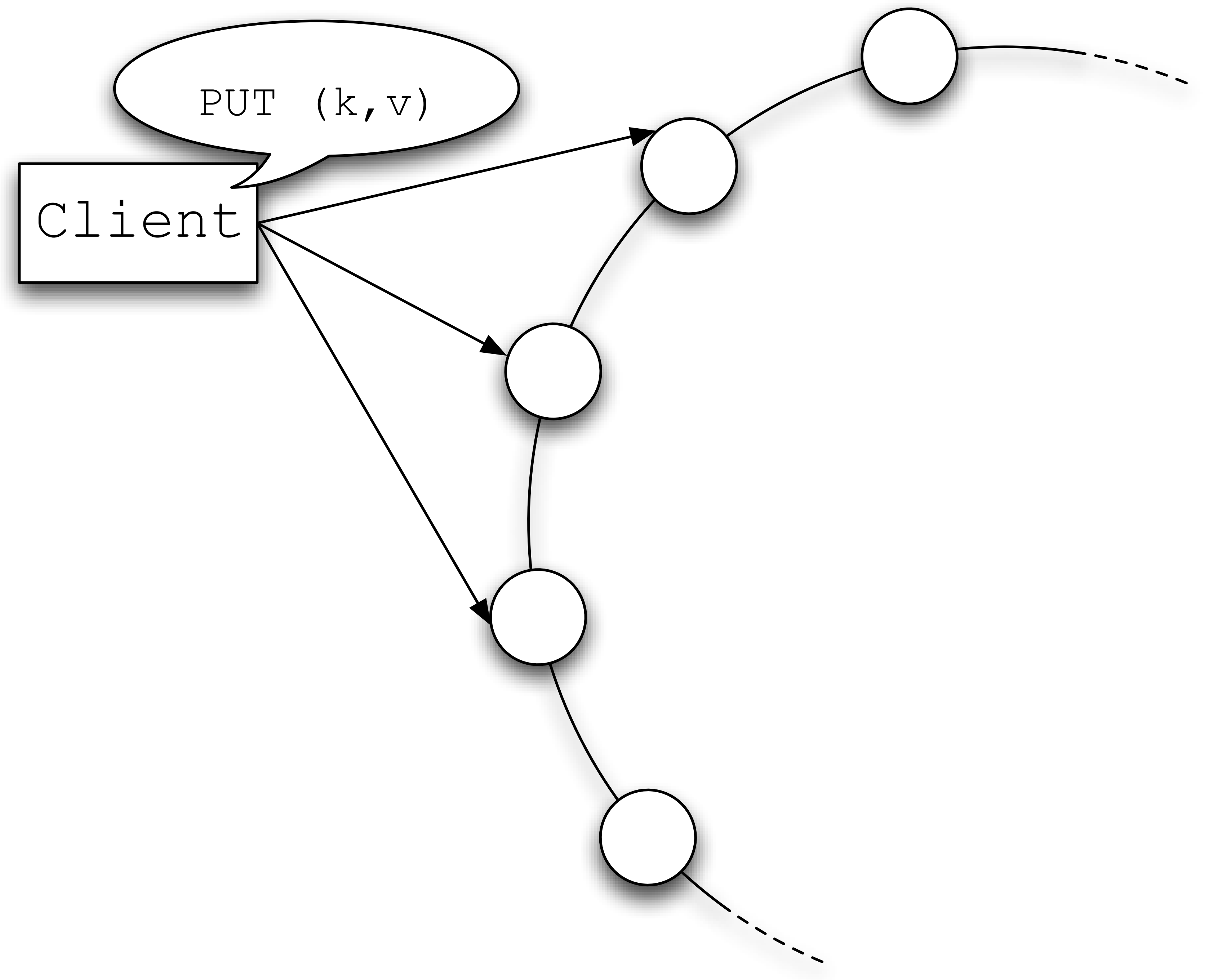
Every Store is Special.

- Lots of different little tweaks to the storage model.
- Widely varying levels of maturity.
- Growing communities.
- Limited (but growing) tooling, libraries, and production adoption.

Reliability Through Replication

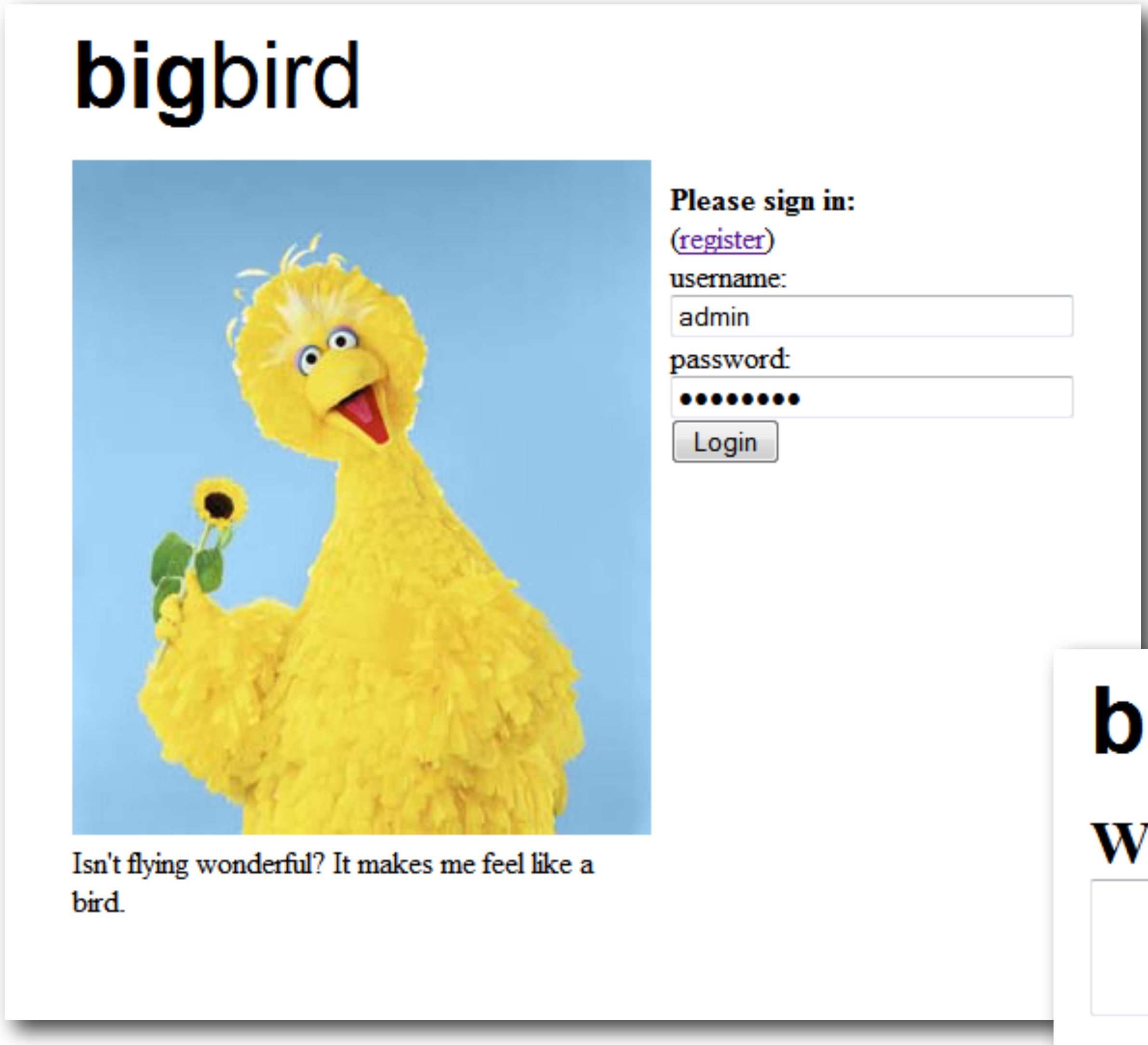
- Consistent hashing to assign keys to partitions.
- Partitions replicated on multiple nodes for redundancy.
- Minimum number of successful reads to consider a write complete.

Reliability Through Replication



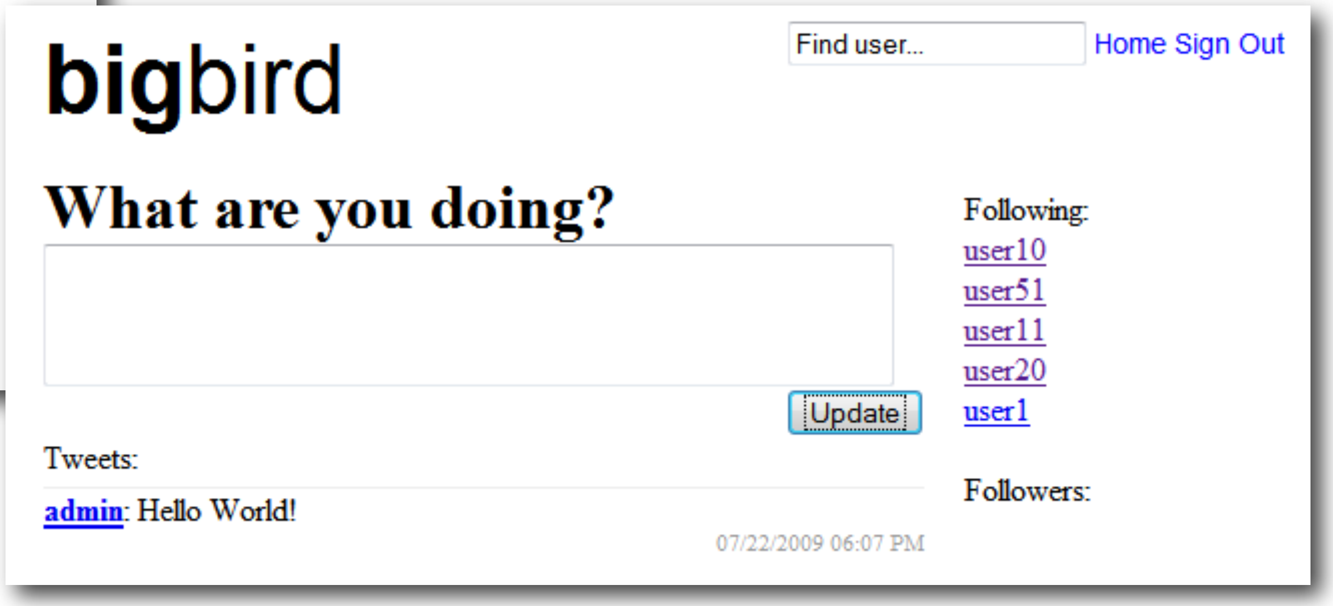
Web UI

http://tat1.datapr0n.com:8080


bigbird

Please sign in:
(register)
username: admin
password: ••••••••

Isn't flying wonderful? It makes me feel like a bird.


bigbird

What are you doing?

Following:
user10
user51
user11
user20
user1

Followers:

Tweets:
admin: Hello World!
07/22/2009 06:07 PM

Stores

- Tweets
  Individual tweets.
- Friends’ Timeline
  Fixed-length timelines.
- Users
  Info and followers.
- Command Queue
  Actions to perform (tweet, follow, etc.).

Data

- Command (Java serialization)

  Keyed by node name, increasing ID.
- Tweets (Java serialization)

  Keyed by user name, increasing ID.
- FriendsTimeline (Java serialization)

  Keyed by username.

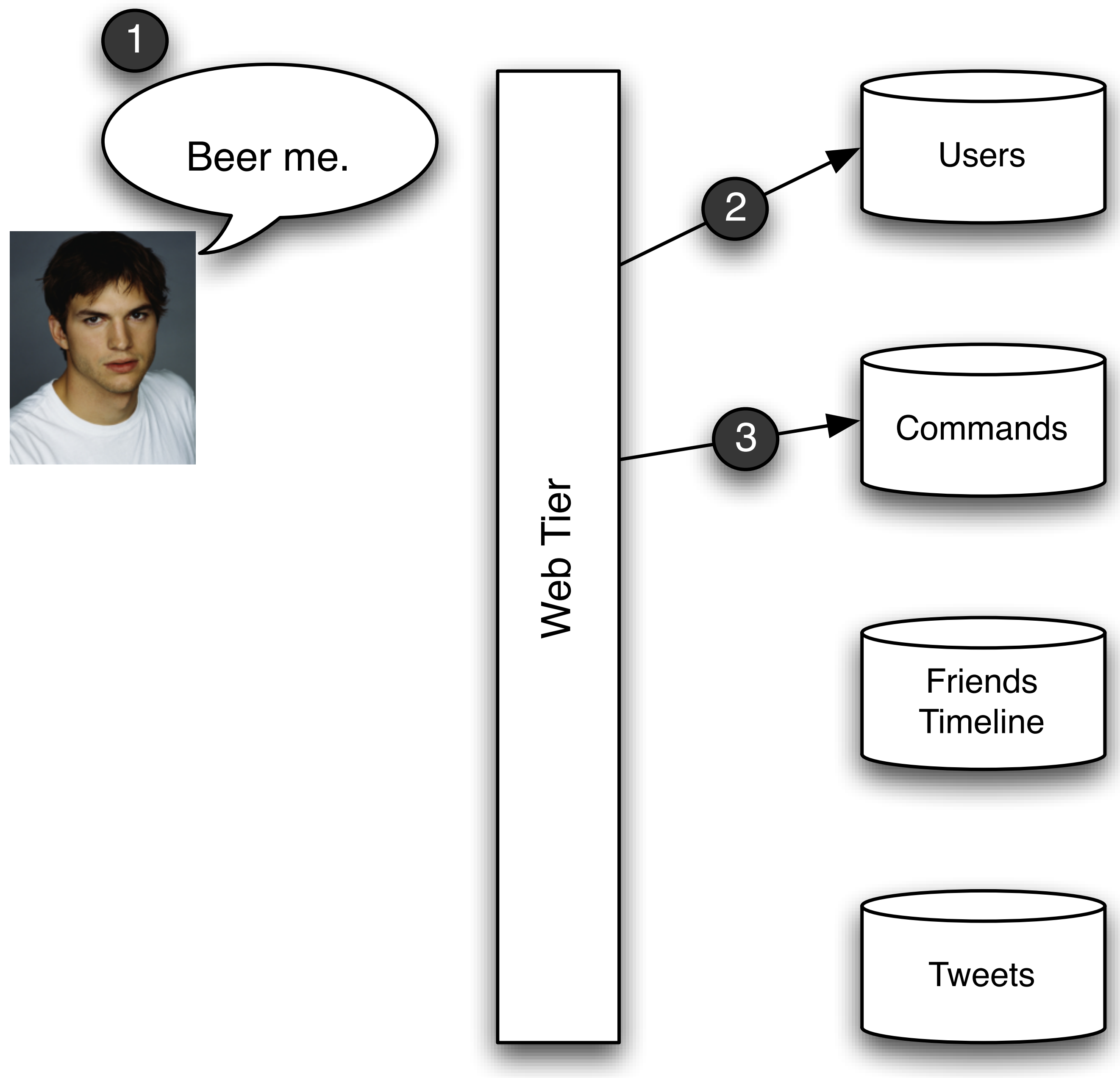
  List of date, tweet ID.
- Users (Java serialization)

  Keyed by username.

  Followers (list), Followed (list), last tweet ID.

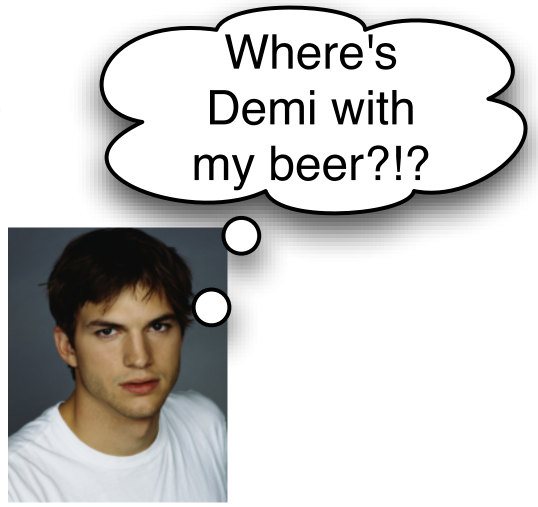
Life of a Tweet, Part I

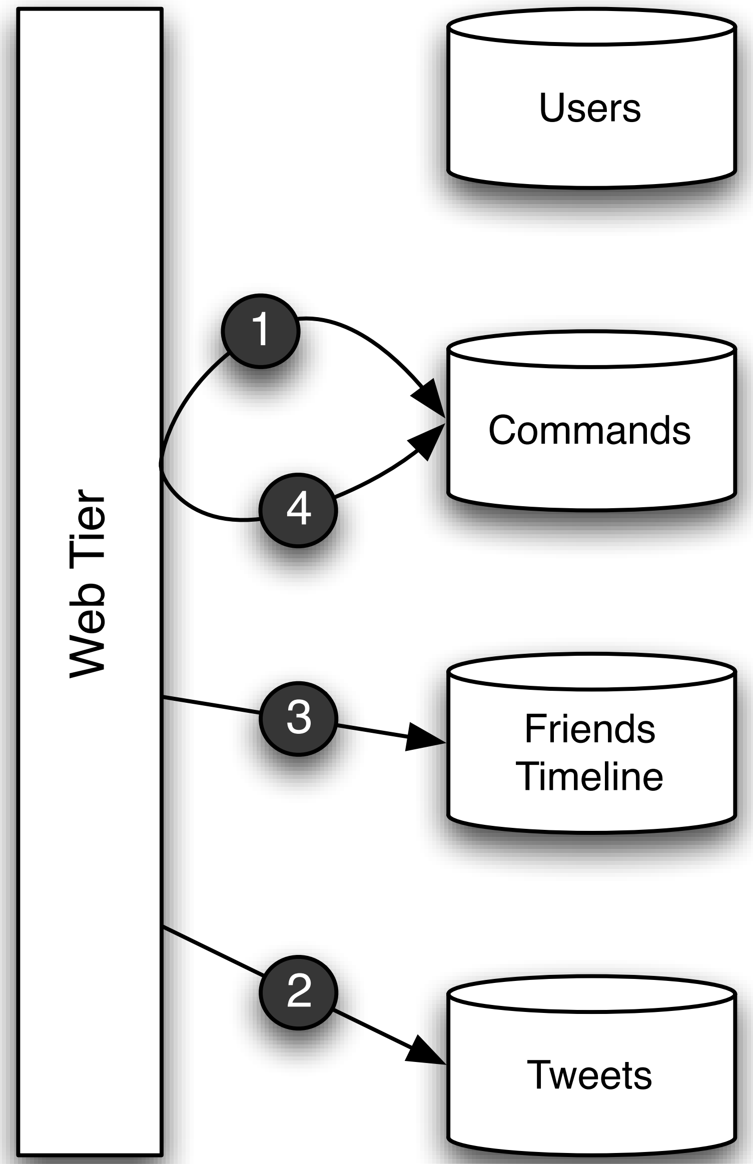
1. User tweets.
2. Find next tweet ID for user.
3. Store "tweet for user" command.



Life of a Tweet, Part II

1. Read next command.
2. Store tweet in user's timeline (Tweets).
3. Store tweet ID in friends' timelines. (Requires *many* operations.)
4. DELETE command.





Some Patterns

- “Sequences” are implemented as race-for-non-collision.
- “Joins” are common keys or keys referenced from values.
- “Transactions” are idempotent operations with DELETE at the end.

Operations

- Deploy to Amazon EC2
  - 2 nodes for Voldemort
  - 2 nodes for Tomcat
  - 1 node for Cacti
- All "small" instances w/RightScale CentOS 5.2 image.
- Minor inconvenience of "EBS" volume for MySQL for Cacti.

  (follow Eric Hammond's tutorial – http://bit.ly/OK5LZ)

Deployment

- Lots of choices for automated rollout (Chef, Capistrano, etc.)
- Took simplest path — Maven build, Ant (scp/ssh and property substitution tasks), and bash scripts.

```
for i in vn1 vn2; do
  ant -Dnode=${i} setup-v-node
done
```

- Takes ~30 seconds to provision a Tomcat or Voldemort node.

Dashboarding

- As above, *lots* of choices

  (Cacti – http://bit.ly/qV4gz, Graphite – http://bit.ly/466NAx, etc.)

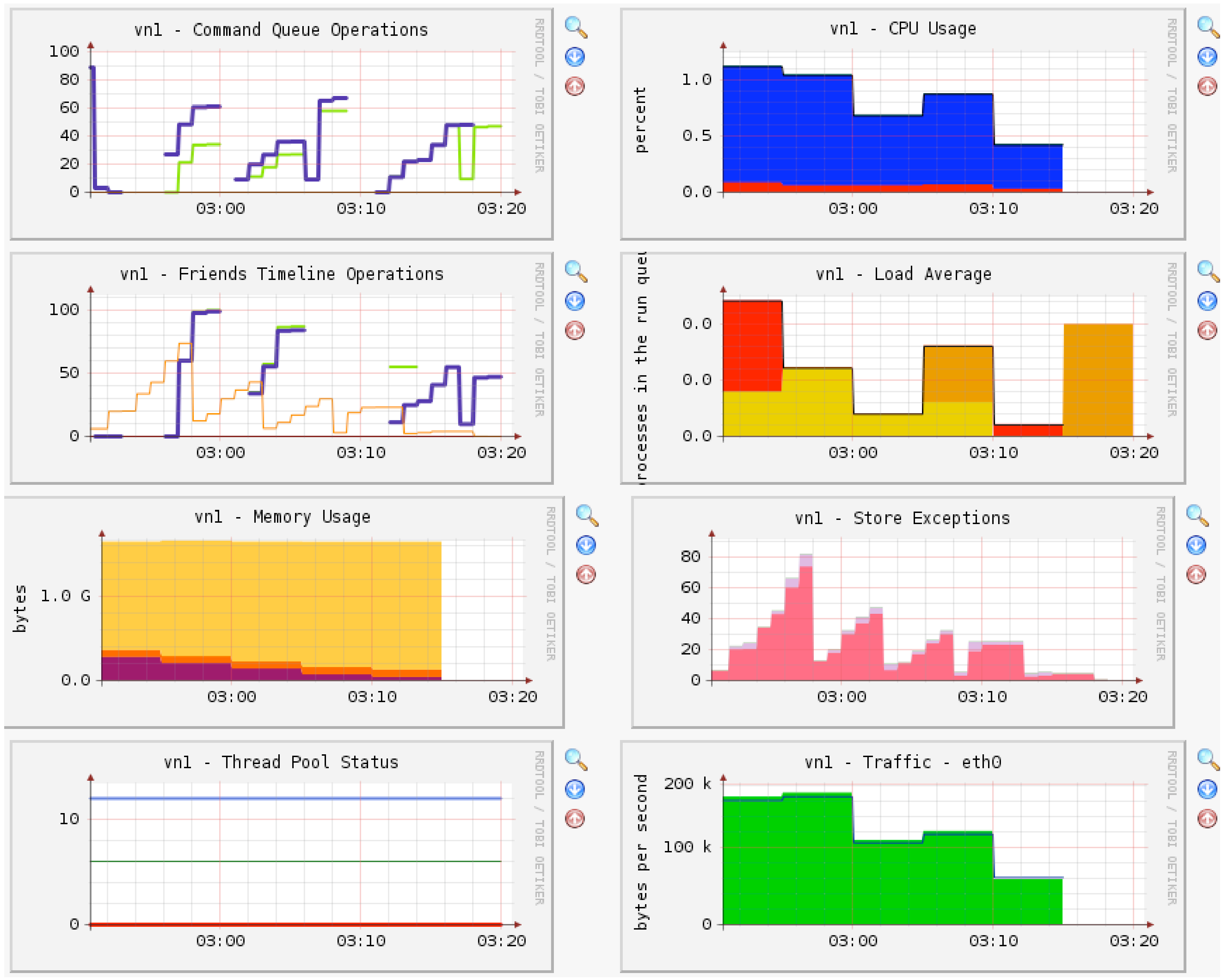
- Cacti as simplest choice.

  ```
  yum install -y cacti
  ```

- Vanilla SNMP on nodes for host data.

- Minimal extensions to Voldemort for stats in Cacti-friendly format.

Dashboarding



Performance

- 270 req/sec for getFriendsTimeline against web tier.
  - 21 GETs on V stores to pull data.
  - 5600 req/sec for V is similar to performance reported at NoSQL meetup (20k req/sec) when adjusted for hardware.
  - Cache on the web tier could make this faster...
- Some hassles when hammering individual keys with rapid updates.

Take Aways

- Linked-list representation deserves some thought (and experiments).

  Dynomite + Osmos (http://bit.ly/BYMdW)

- Additional use cases (search, rich API, replies, direct messages, etc.) might alter design.
- BigTable/HBase approach deserves another look.
- Source code is available; come and git it.

  http://github.com/prb/bigbird

  git://github.com/prb/bigbird.git

Coordinates

- Dan Diephouse (@dandiep)

  dan@netzooid.com

  http://netzooid.com
- Paul Brown (@paulrbrown)

  prb@mult.ifario.us

  http://mult.ifario.us/a