Step 3: Prophet

clkao@bestpractical.com

BEST PRACTICAL ™

Prophet

A semirelational,
peer to peer replicated,
disconnected, versioned,
property database with
self-healing conflict resolution

BEST PRACTICAL ™

What do all those *buzzwords* mean?

BEST PRACTICAL ™

semirelational

Joins are expensive

Joins are discouraged

BEST PRACTICAL ™

peer-to-peer replicated

Update any replica

Pull from any replica

Push *to* any replica

Publish a replica

Changes will propagate

BEST PRACTICAL ™

disconnected

Real-time replication is hard to scale

It only works in the cloud

I don't live in the cloud

(Except when I go hiking)

I want my data when I'm offline

Prophet sync can happen whenever

versioned

Compare a record to any point in the past

All changes fully logged

Undo changes

Use history to be smart

BEST PRACTICAL ™

property database

Atomic changes

CREATE, READ, UPDATE, DELETE, SEARCH

Record typess can have optional validation and canonicalization

Records of the same type do not need to have the same properties

Add and remove properties at will

BEST PRACTICAL ™

self-healing conflict resolution

- Prophet remembers all conflict resolutions
- Prophet detects identical conflicts
- Prophet can use your peers' resolutions as "votes" for the winner in a conflict

BEST PRACTICAL ™

What could you build with Prophet?

BEST PRACTICAL ™

sd

A bug tracker: “simple defects”

BEST PRACTICAL ™

```
./bin/sd ticket create
   --summary "Can't sync sd with Google Code"
   --status new


Created ticket 93BF979E-08C1-11DD-94C3-D4B1FCEE7EC4
```

BEST PRACTICAL ™

```
./bin/sd ticket search --regex .


93BF979E-08C1-11DD-94C3-D4B1FCEE7EC4 Can't sync sd
with Google Code new
```

```
./bin/sd ticket update
  --uuid 93BF979E-08C1-11DD-94C3-D4B1FCEE7EC4
  --status resolved


./bin/sd ticket search --regex .
93BF979E-08C1-11DD-94C3-D4B1FCEE7EC4 Can't sync sd
with Google Code resolved
```

BEST PRACTICAL ™

Bugs on my laptop alone aren't interesting.

BEST PRACTICAL ™

Syncable!

```
./bin/sd publish /tmp/mybugs

scp -rvp /tmp/mybugs myserver.com:public_html/sd
```

Bob:

```
./bin/sd pull prophet:http://myserver.com/~clkao/sd
```

Same thing over svn

(svn also supports push)

My project has a bug tracker

Actually, mine use two:

RT

hiveminder.com

BEST PRACTICAL ™

Wrote an RT Replica for SD

(Using only the public REST API)

It took an afternoon

Mirror an RT instance into SD

Share it with your peers using prophet

Sync changes back from your peers to RT

...and one for Hiveminder

(Using only the public REST API)

I can sync my bugs with RT or Hiveminder


Actually, it's better
BEST PRACTICAL ™

I can sync between RT *and* Hiveminder

BEST PRACTICAL ™

I can sync between two different RTs, too

BEST PRACTICAL ™

We need more replica definitions:

- Trac
- Google Code
- SourceForge
- Bugzilla
- Jira
- GForge
- debbugs
- GNATS
- What else?

BEST PRACTICAL ™

What should I use Prophet for?

All your “little” databases

BEST PRACTICAL ™

All the databases you want while offline.

- CRM
- Bug tracking
- Sales orders
- Phone book
- Blog
- Trading Card Database
- Ideas?

BEST PRACTICAL ™

A look inside Prophet

Terms and Concepts

- Database
- Replica
- Foreign Replica
- Record Type (or just 'Type')
- Record
- Change Set
- Change
- Property Change
- Conflict
- Nullifcation Change Set
- Resolution Change Set
- Merge
- Merge Ticket

BEST PRACTICAL ™

Replica Backends

The Record Store

Stores indivdual records by type

The Changeset Store

Stores each atomic change to a set of records

Replaying all changesets will create an exact clone of the replica

BEST PRACTICAL ™

Native Replica Types

Subversion

- Slow
- Steady
- Robust
- Supports remote sync
- Requires Subversion Perl Bindings

Filesystem

Readable

Flat files

Compact

Fast

Currently export-only. (Updateable soon!)

HTTP

Flat-files, Currently read-only.

Same format as the filesystem replica type.

“Foriegn” replicas

Will usually be app specific

All current examples are in SD

Synchronization

Publish

Serialize and export all of a replica's resolutions and changesets

Pull

Integrate unseen resolutions and then unseen changesets from a replica

Push

Integrate new resolutions and changesets into a replica

Resolving Conflicts

Figures out the best resolution

"Nullifies" the conflict so the changeset can be cleanly integrated

Integrates the conflicting changeset

Records the resolution as a new changeset

Records the resolution decision in the resolution database

“The Best Resolution”

Prophet has clever ways to figure out the best resolution.

If there are previous resolutions for the same conflict *and* a majority agree, use that

If the merger has specified a “prefer this side” choice, use that

Prompt the user to make a decision, giving them info about previous decisions for this conflict

BEST PRACTICAL ™

We don't have a proof

We *do* have dozens of runs of randomized testing.

So far, it always stabilizes sanely.

How does it scale?

Vertical scale is boring

Designed to scale to many peers

You are not Google

(Most of you are not Google. Hi, Googlers!)

Current target is databases of O(50k) records

Why not, then?

We just have a political agenda.

Web 2.0 is not Open Source.

Your data should not be ‘exportable’ from the cloud.

You should **always** have full control.

So we dont need to store 10 billion records in one database.

BEST PRACTICAL ™

(Do you have 10 billion bugs, customer contacts or sales orders?)

BEST PRACTICAL ™

That said, we'd love to see a scalable, high performance prophet replica store

BEST PRACTICAL ™

Our Plans

Search Indexes

Full Text Search

Query language

Non-subversion native replica format

Enhanced REST Server

Jifty, Catalyst, Rails models?

BEST PRACTICAL ™

Project Metadata

Prophet is *very young*

Many, many hours of design

About 10 days of two hackers hacking

Codebase

Prophet

4837 lines of code and doc

1455 lines of tests

sd

1479 lines of code and doc

446 lines of tests

BEST PRACTICAL ™

Getting Prophet

Prophet

http://code.bestpractical.com/bps-public/Prophet/

SD

http://code.bestpractical.com/bps-public/sd/

Getting Involved

prophet-subscribe@lists.bestpractical.com

Thanks!

BEST PRACTICAL ™