Step 3: Prophet

jesse@bestpractical.com
clkao@bestpractical.com

BEST PRACTICAL ™

Cloud computing is Sharecropping.

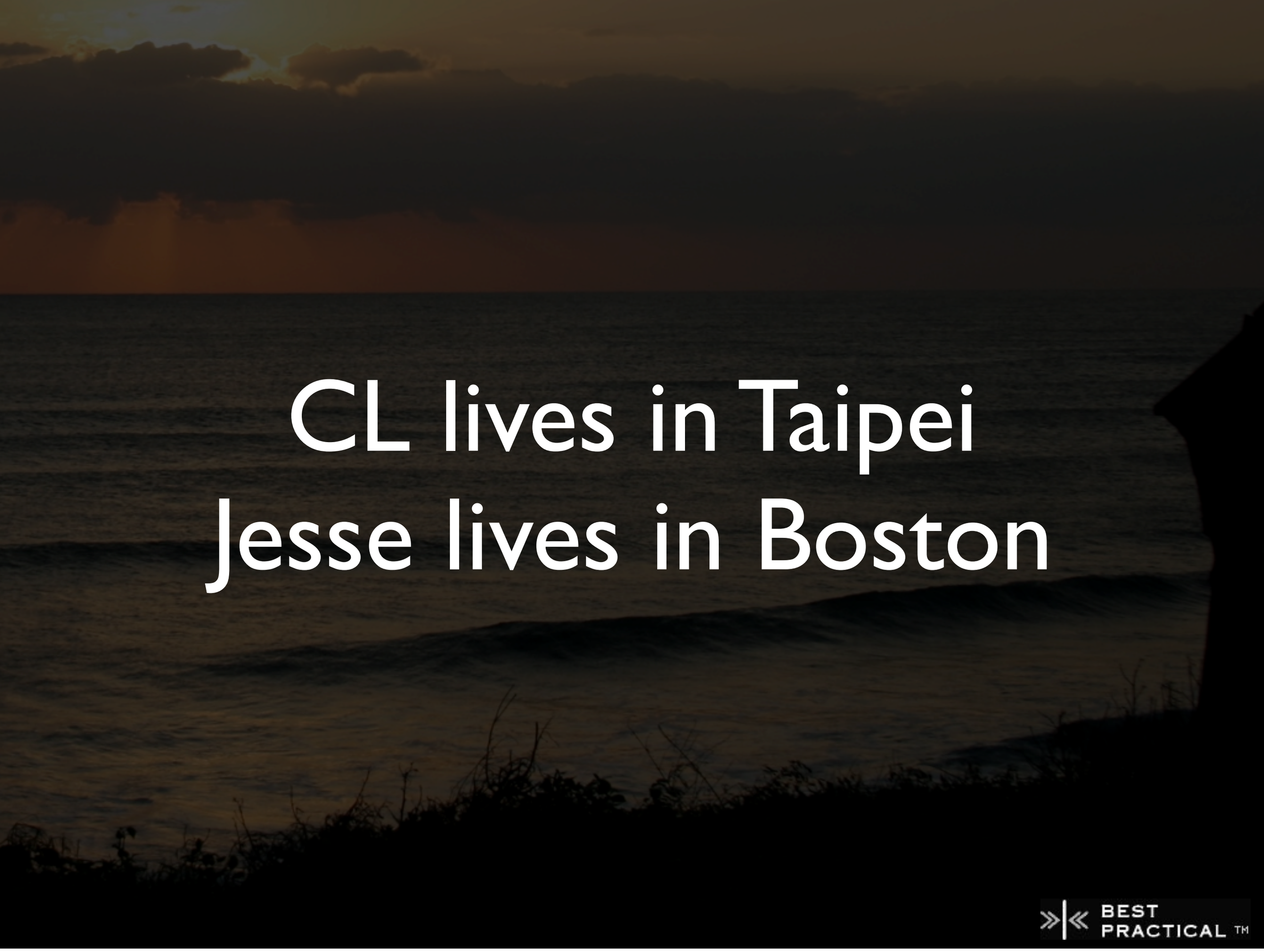
(小作農)

BEST PRACTICAL ™


Chia-liang Kao
BEST PRACTICAL ™


Jesse Vincent
BEST PRACTICAL ™


We work together
BEST PRACTICAL ™


CL lives in Taipei
Jesse lives in Boston
BEST PRACTICAL ™

Sometimes we need to work face to face

BEST PRACTICAL ™

TPE~BOS: 9410 mi

TPE-HNL: 5,095 mi

BOS-HNL: 5,069 mi

Our Plan

Step 1: Go to Hawaii for "work"

Step 2: ???

Step 3: Prophet!

The Plan Backfired

We were there for 8 days

We wrote 8000 lines of Perl

We figured out step 2

Step 2:

Build a Disconnected Syncable Database

BEST PRACTICAL ™

Prophet

Prophet

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

BEST PRACTICAL ™

Bribery!

Two tshirts.

If you write a (real) patch for Prophet during this talk, you get a tshirt.

Send patches to jesse@bestpractical.com

Getting Prophet

Prophet

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

SD

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

What do all those *buzzwords* mean?

BEST PRACTICAL ™

semirelational

Joins are expensive

No backend Join support (yet)

peer-to-peer replicated

Update any replica

Pull from any replica

Push *to* any replica

Publish a replica

Changes will propagate

disconnected

Real-time replication is hard to scale

It only works in the cloud

I don't live in the cloud

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 operations

CREATE, READ, UPDATE, DELETE, SEARCH

Record types 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

Remembers all conflict resolutions

Syncs all resolutions with your peers

Detects identical conflicts

Uses your peers' resolutions to "vote" for the winner of a conflict

BEST PRACTICAL ™

What could you build with Prophet?

BEST PRACTICAL ™

sd

A bug tracker: “simple defects”

- id. Status, Summary
- History
- Comments
- Attachments

BEST PRACTICAL ™

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


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

```
./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 aren't interesting.

BEST PRACTICAL ™

Syncable!

Jesse

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

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

CL

```
./bin/sd pull prophet:http://my.com/~jesse/sd
```

My project has a bug tracker

BEST PRACTICAL ™

My project has a bug tracker

Actually, mine use two:

- RT
- hiveminder.com

Foreign Replicas

Prophet makes Foriegn Replicas easy

SD gets them "for free"

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

Supports Comments and Attachments

BEST PRACTICAL ™

...and one for Hiveminder

(Using only the public REST API)

I can sync my bugs with RT or Hiveminder

BEST PRACTICAL ™

Actually, it's better

I can sync between RT *and* Hiveminder

BEST PRACTICAL ™

I can sync between two different RTs, too

We need more replica definitions:

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

What can you use Prophet for?

BEST PRACTICAL ™

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 ™

“Private” Social Networks

How about a P2P BBS?

Prophet doesn’t need a server.

You can sync over sneakernet.

BEST PRACTICAL ™

A look inside Prophet

Terms and Concepts

- Database
- Database Replica
- “Foreign” Replica
- Record
- Record Type
- Change Set
- Change
- Property Change
- Conflict
- Resolution
- Nullifcation Change Set
- Resolution Change Set
- Merge
- Merge Ticket

BEST PRACTICAL ™

Database 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

Native Replica Types

Subversion

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

BEST PRACTICAL ™

Filesystem

Readable

Flat files

Compact

Fast

(Not yet fully atomic)

HTTP

Designed to let you “publish” databases

Flat-files, Currently read-only.

Same format as the filesystem replica type.

“Foreign” replicas

Will usually be app specific

All current examples are in SD

BEST PRACTICAL ™

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

BEST PRACTICAL ™

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

BEST PRACTICAL ™

“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 for the algorithm yet

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

So far, it always stabilizes sanely.

BEST PRACTICAL ™

How does it scale?

Vertical scale is boring

Designed to scale to many peers

You are not Google

Current target is databases of O(50k) records

BEST PRACTICAL ™

How does it scale?

Vertical scale is boring

Designed to scale to many peers

You are not Google

Does anyone here work for Google?

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 shouldn't 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 ™

Project Status

Simple, well-defined Perl API

RESTy web API (with microserver)

Fast, lightweight backend

Small, active dev community

Great test coverage

Horrible POD coverage

BEST PRACTICAL ™

Our Plans

Improved search and indexing

(Including full-text indexing)

Query language

Proper security model

Jifty, Catalyst, Rails models?

BEST PRACTICAL ™

Prophet is *very young*

Many, many hours of design

About 10 days of two hackers hacking

BEST PRACTICAL ™

Codebase

Prophet

5479 lines of code and doc

1693 lines of tests

sd

1695 lines of code and doc

876 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

#prophet on freenode IRC

Thanks!

BEST PRACTICAL ™