WHERE TO PUT DATA

—or—

What are we going to do with all this stuff?
About The Speaker

Application Developer/Architect – 20 years
Web Developer – 15 years
IT Operations – 7 years
IT Services Executive - 6 years
Relational

Shard

Replicate

Document

Key-value

NetStorage

S3

CloudFront

ACID

Relational

Cluster

Coherence

NetStorage

S3

CloudFront

ACID

Relational

Cluster

Coherence

MongoDB

Distributed

HBase

Shard

Schemaless

Document

Cassandra

Cache

CouchDB

Schema

HDFS

BigMemory

Redis

Graph

Neo4J

BASE

Xindice

eXist

GridGain

Riak

Replicate

Cluster

Coherence

Memcached

BDB

BDB

Voldemort

Voldemort

Voldemort

Voldemort

Voldemort

Voldemort
BACK IN THE 90’S
BACK IN THE 90’S
Hierarchical ("Network") Database

OS 2200
OS 2200

Hierarchical ("Network") Database

Relational Mapper

POSIX.1 Virtual Machine

OS 2200
OS 2200

Hierarchical ("Network") Database

Relational Mapper

ANSI SQL Library

POSIX.1 Virtual Machine

COBOL Compiler
Given enough time, and perversity, you can create any query model on top of any storage model.
FUNDAMENTAL PREMISE

Black Box
THERE ARE THINGS YOU CANNOT KNOW

Will a response arrive?

When?

Was it stored or computed?

Is it still true?
SAY WHEN
The Importance of Response Time Distribution
ASYMMETRY OF TIME

Send Request
ASYMMETRY OF TIME

Send Request

100 ms 200 ms
ASYMMETRY OF TIME

Send Request

100 ms 200 ms 300 ms 400 ms 500 ms 600 ms

Wednesday, October 6, 2010
ASYMMETRY OF TIME
ASYMMETRY OF TIME
To the observer, there is no difference between “too slow” and “not there”.

Wednesday, October 6, 2010
Response Time Histogram

% of Responses

Response Time (ms)

Wednesday, October 6, 2010
Response Time Histogram

% of Responses

Response Time (ms)

Wednesday, October 6, 2010
This is a talk about data and data storage.

Why am I talking so much about observers and response time?

What about scalability?
Why do we worry about scalability?
Empty Box Time
Response time of a single request on an unloaded system.

Wednesday, October 6, 2010
Scalability is a proxy.

What we want is fast responses, under all loads.
HOW TRUE?
SAYS WHO?

Thoughts on Consistency
OBSERVABILITY
OBSERVABILITY

Steve

Brian
OBSERVABILITY
OBSERVABILITY

<table>
<thead>
<tr>
<th>#</th>
<th>Steve</th>
<th>Brian</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>L</td>
<td>R</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>L</td>
</tr>
</tbody>
</table>

Steve

Brian

Wednesday, October 6, 2010
STATE SPACE

\[ X_1 = \{L, R\} \]

\[ X_2 = \{L, R\} \]
SUPER-OBSERVER

Has a view which *dominates* the views of all other observers.
SUPER-OBSERVER

There are no one-to-many mappings from the super-observer’s states to any other observer’s states.
A super-observer is maximally present if it can discriminate among the Cartesian product of all other observations.

<table>
<thead>
<tr>
<th>Observer</th>
<th>Set of States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steve</td>
<td>{L, R}</td>
</tr>
<tr>
<td>Brian</td>
<td>{L, R}</td>
</tr>
<tr>
<td>Super-Observable</td>
<td>{L → B, R → F} × {L, R}</td>
</tr>
</tbody>
</table>
# Observability

<table>
<thead>
<tr>
<th>#</th>
<th>Steve</th>
<th>Brian</th>
<th>Super-Observer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>R</td>
<td>{F, R}</td>
</tr>
<tr>
<td>2</td>
<td>L</td>
<td>R</td>
<td>{B, R}</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>L</td>
<td>{F, L}</td>
</tr>
</tbody>
</table>
STATE SPACE

Cartesian product of all possible sets of states.

Example
1,000,000 bytes of RAM
8 bits per byte
2 states per bit

8,000,000 dimensions with 2 values each
or
1,000,000 dimensions with 256 values each
STATE SPACE

10,000,000 rows in a table
20 columns

Whole database is a single point in a 200,000,000 dimensional space.

Changes to data are transforms of that point.

State over time is the trajectory of that point.
CONSISTENCY

Not every point in state space is allowed.
PORKY PIG
PORKY PIG
FORBIDDEN STATES

$X_2 = \{\text{shade open, shade closed}\}$

$X_1 = \{\text{looking, not looking}\}$
BLACK BOX HYPOTHESIS
BLACK BOX HYPOTHESIS

External observers can only ever ask for projections of the state space, at defined points in time.
BLACK BOX HYPOTHESIS

State space trajectories may cross into forbidden states, as long as those are not revealed to observers.
Is Porky looking at the window shade?

\[ X_1 = \{\text{looking, not looking}\} \]
Even two clustered machines have their own state spaces.

It’s impossible for either to be a superobserver.
OBSERVED CONSISTENCY

Sufficient to ensure that forbidden states cannot be observed.
DOES A SUPEROBSERVER EXIST?

Only if there is exactly one single-threaded CPU, in exactly one computer.
CONS KE NCE S

Consistency doesn’t exist in most systems today.

Sometimes we can fake it.

Many times, it doesn’t really matter.
WHAT ABOUT CAP?

Consistency:

“...there must exist a total order on all operations such that each operation looks as if it were completed at a single instant.”

Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services.

*SIGACT News* 33, 2 (June 2002), 51-59. DOI=10.1145/564585.564601
http://doi.acm.org/10.1145/564585.564601
WHAT ABOUT CAP?

Linearizability


SIGACT News 33, 2 (June 2002), 51-59. DOI=10.1145/564585.564601
http://doi.acm.org/10.1145/564585.564601
CONSISTENCY IS NOT BOOLEAN

Eventual
Monotonic read
Read-your-own-writes
Monotonic + Read-your-own-writes
Immediate
Strong
Full transactions

From the MongoDB Blog

Wednesday, October 6, 2010
“C” VERSUS “A”? 

Resilience ← Response Time → Consistency

See also: http://goo.gl/1Yv3
HOW LONG?
On Lifecycles and Lifespans
DOWN WITH THE IRON FIST

Throw out the DBAs
Throw out the schemas
Unstructured
Semi-structured
DOWN WITH THE IRON FIST

Put the application in charge.
DOWN WITH THE IRON FIST

but...
DIFFICULTIES

Application versions

Validating correct behavior

Capturing knowledge about that behavior
UGLY TRUTH

Data routinely outlives applications.
WHERE NOW?
WHERE TO PUT YOUR DATA?

Data exists everywhere.
WHERE TO PUT YOUR DATA?

Nothing lasts forever.
WHERE TO PUT YOUR DATA?

Understand freshness.
WHERE TO PUT YOUR DATA?

Engineer a good response time distribution.
WHERE TO PUT YOUR DATA?

Select the consistency model you need.
WHERE TO PUT YOUR DATA?

Be agile and adaptable.
WHERE TO PUT YOUR DATA?

Make it sustainable.