Soul-searching For The Relational Camp

Why NoSQL and Big Data Have Momentum

NoSQL “Gangnam Style”

http://www.youtube.com/watch?v=fXc-QDJBXpw

Objectives

- The origins of NoSQL
- The NoSQL product landscape
- *What* makes Relational so sacred?
- The mistakes of the relational camp
- Learning resources
- Bonus questions
  - Performance, scalability, and reliability without NoSQL?
  - What about that CAP theorem?
  - Should I study NoSQL?

Non-Objectives

- Installation and operation of NoSQL products

Soul-searching For The Relational Camp

- The Origins of NoSQL

Dynamo Requirements

1. Customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados. Therefore, the service responsible for managing shopping carts requires that it can always write to and read from its data store, and that its data needs to be available across multiple data centers.

Dynamo Requirements (Contd.)

2. There are many services on Amazon's platform that only need primary-key access to a data store. For many services, such as those that provide best seller lists, shopping carts, customer preferences, session management, sales rank, and product catalog, the common pattern of using a relational database would lead to inefficiencies and limit scale and availability. Dynamo provides a simple primary-key only interface to meet the requirements of these applications.
3. Since each service uses its distinct instance of Dynamo, its initial design targets a scale of up to hundreds of storage hosts [only].

Dynamo Requirements (Contd.)

3. Experience at Amazon has shown that data stores that provide ACID guarantees tend to have poor availability. … Dynamo targets applications that operate with weaker consistency (the “C” in ACID) if this results in high availability.
4. Dynamo is used only by Amazon’s internal services. Its operation environment is assumed to be non-hostile and there are no security related requirements such as authentication and authorization.

Dynamo Requirements (Summary)

1. Extreme performance
2. Extreme scalability
3. Extreme availability

Dynamo Solution—Functional Segmentation

- Best seller lists, shopping carts, customer preferences, session management, sales rank, and product catalog
- Increases overall site availability by avoiding a single point of failure
- No distributed transactions

Dynamo Solution—Sharding

- **employee** (employee#, name, birthdate)
- **jobhistory** (employee#, jobdate, title)
- **salaryhistory** (employee#, jobdate, salarydate, salary)
- **children** (employee#, childname, birthyear)

Dynamo Solution—Replication

- Multi-master replication
- Eventual consistency

Dynamo Solution—BLOBs

- Shopping carts are stored as binary objects (i.e., blobs) identified by unique keys. No operations span multiple data items and there is no need for relational schema.
- *"Using tables to store objects is like driving your car home and then disassembling it to put it in the garage. It can be assembled again in the morning, but one eventually asks whether this is the most efficient way to park a car"*—attributed to Esther Dyson

Dynamo Solution—Summary

- Functional segmentation
- Sharding
- Asynchronous replication
- BLOBs
- No SQL
- Primary-key access
- Autocommit
- No distributed transactions
- Eventual consistency

eBay Solution

- Oracle
- SQL
- Functional segmentation
- Sharding
- Local transactions with ACID
- No distributed transactions
- Middle-tier caching
- Middle-tier constraint checking
- Asynchronous replication for search

Codd on Eventual Consistency

- “There are, of course, several possible ways in which a system can detect inconsistencies and respond to them. In one approach the system checks for possible inconsistency whenever an insertion, deletion, or key update occurs. **Naturally, such checking will slow these operations down.** If an inconsistency has been generated, details are logged internally, and if it is not remedied within some reasonable time interval, either the user or someone responsible for the security and integrity of the data is notified. Another approach is to conduct consistency checking as a batch operation once a day or less frequently” — Codd, E. F. “A relational model of data for large shared data banks.” (1970).

Zeroth Normal Form

- *"Nonatomic values can be discussed within the relational framework. Thus, some domains may have relations as elements. These relations may, in turn, be defined on nonsimple domains, and so on. For example, one of the domains on which the relation employee is defined might be salary history."*
- **employee** (
  employee#,
  name,
  birthdate,
  jobhistory (jobdate, title, **salaryhistory** (salarydate, salary)),
  children (childname, birthyear)
)

First Normal Form

- employee' (employee#, name, birthdate)
- jobhistory' (employee#, jobdate, title)
- salaryhistory' (employee#, jobdate, salarydate, salary)
- children' (employee#, childname, birthyear)

Oracle Table Clusters

- Data from multiple tables stored in the same block
- Hash clusters
- Indexed clusters
- Clustered tables can be indexed

Soul-searching For The Relational Camp

- Oracle Table Clusters—Demonstration
  Refer to clusters.sql and clusters.log
  Insructions available at
  http://iggyfernandez.wordpress.com/2013/07/28/no-to-sql-and-no-to-nosql/

Soul-searching For The Relational Camp

- The NoSQL Landscape

The NoSQL Landscape

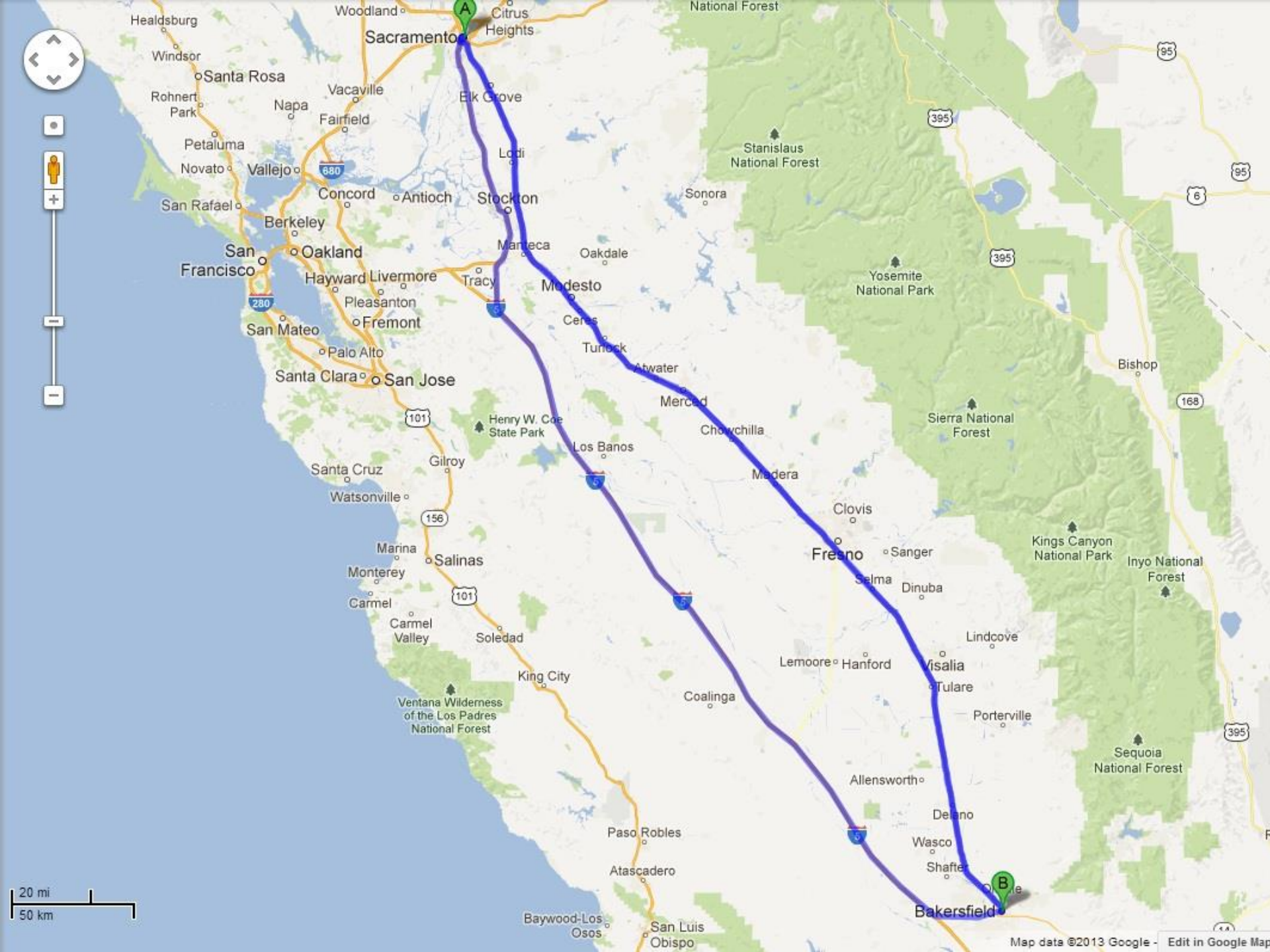
- Key-Value Stores—Dynamo, Riak, Oracle NoSQL
- Document Stores—Mongodb
- Column Stores—Cassandra, HBase
- Graph Databases—Neo4J
- Big Data—Hadoop

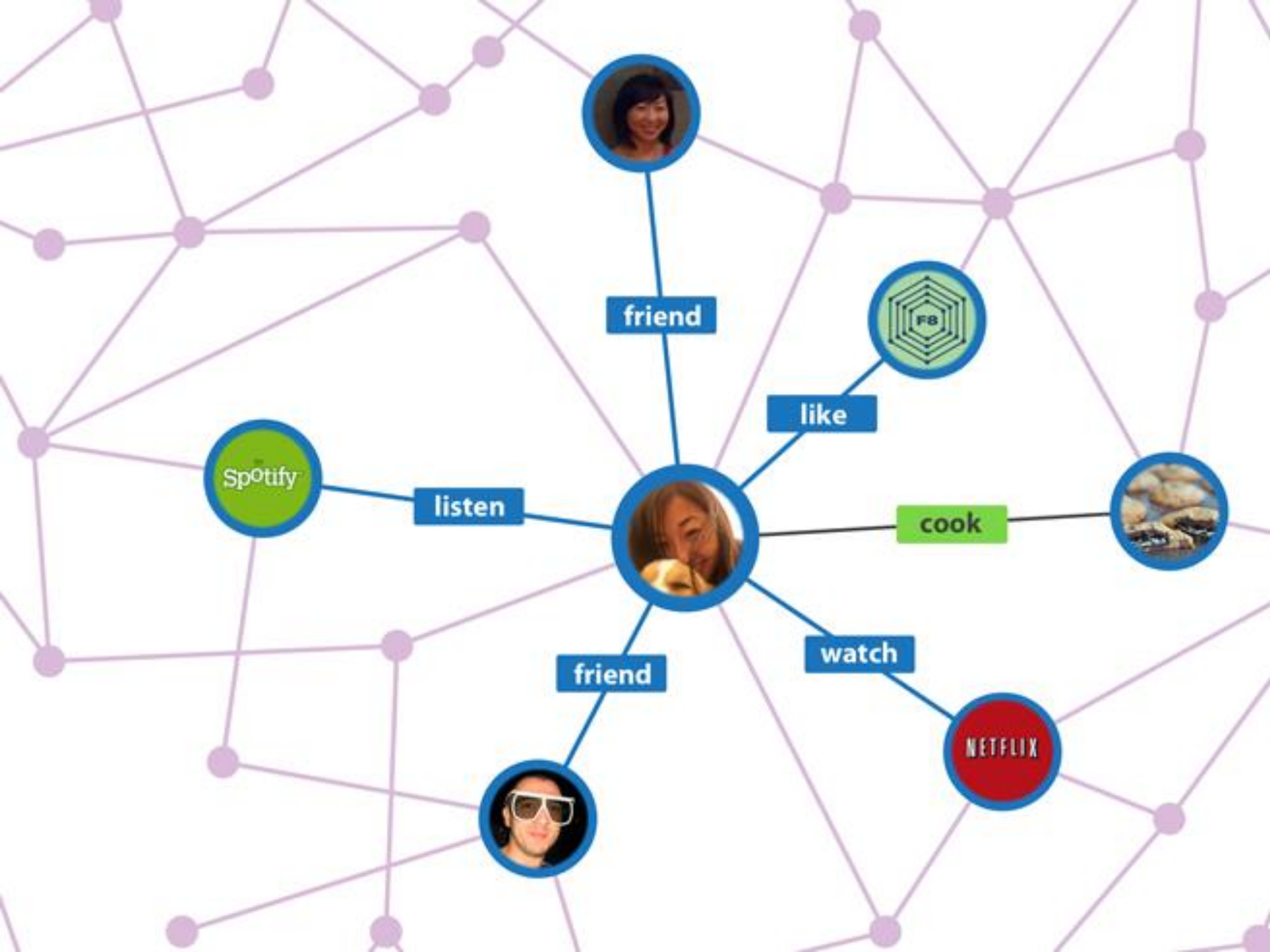
Document Store—MongoDB


```
{
    "_id": ObjectId("4efa8d2b7d284dad101e4bc9"),
    "Last Name": "DUMONT",
    "First Name": "Jean",
    "Date of Birth": "01-22-1963"
},
{
    "_id": ObjectId("4efa8d2b7d284dad101e4bc7"),
    "Last Name": "PELLERIN",
    "First Name": "Franck",
    "Date of Birth": "09-19-1983",
    "Address": "1 chemin des Loges",
    "City": "VERSAILLES"
}
```

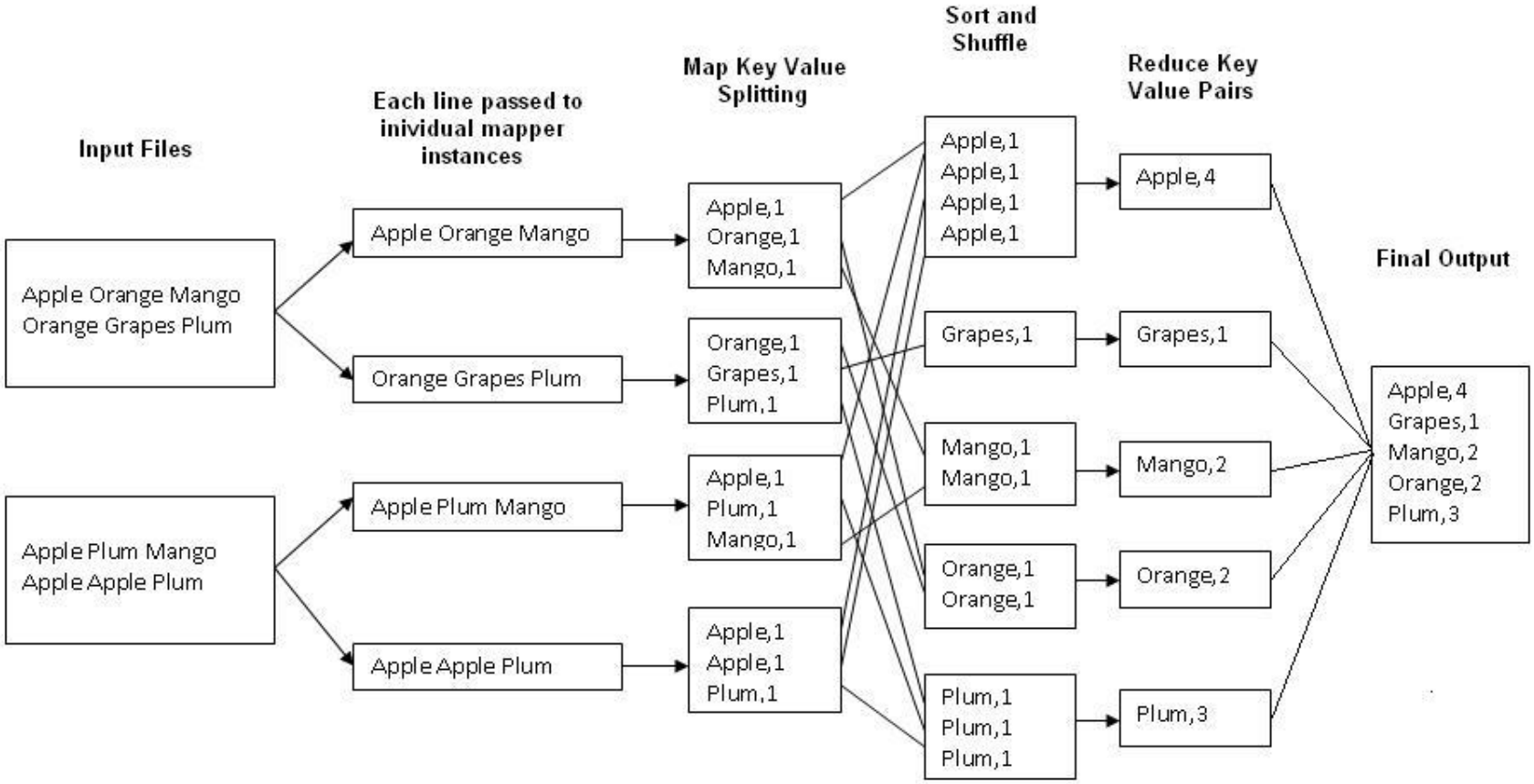
MongoDB

```
{
    "_id":
     ObjectId("4efa8d2b7d284dad101e4bc7"),
    "Last Name": "PELLERIN",
    "First Name": "Franck",
    "Date of Birth": "09-19-1983",
    "phoneNumber": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567",
            "verified":  false
        }
    ],
    "Address": {
        "Street": "1 chemin des Loges",
        "City": "VERSAILLES"
    },
    "Months at Present Address":  37
}
```


A
B
Sacramento
Citrus Heights
Elk Grove
Lodi
Stockton
Manteca
Tracy
Oakdale
Modesto
Ceres
Turlock
Atwater
Merced
Chowchilla
Madera
Fresno
Clovis
Sanger
Selma
Dinuba
Kings Canyon National Park
Inyo National Forest
Sierra National Forest
Yosemite National Park
Stanislaus National Forest
Sonora
Bishop
Lindcove
Lemoore
Hanford
Visalia
Tulare
Porterville
Sequoia National Forest
Allensworth
Delano
Wasco
Shafter
Bakersfield
Coalinga
King City
Paso Robles
Atascadero
Baywood-Los Osos
San Luis Obispo
Los Banos
Henry W. Coe State Park
Gilroy
Santa Cruz
Watsonville
Marina
Salinas
Monterey
Carmel
Carmel Valley
Soledad
Ventana Wilderness of the Los Padres National Forest
Santa Clara
San Jose
Palo Alto
San Mateo
Fremont
Hayward
Livermore
Pleasanton
San Francisco
Oakland
Berkeley
San Rafael
Concord
Antioch
Vallejo
Novato
Petaluma
Rohnert Park
Santa Rosa
Windsor
Healdsburg
Woodland
Vacaville
Fairfield
Napa
National Forest
20 mi
50 km
Map data ©2013 Google -
Edit in Google Map


friend
F8
like
Spotify
listen
cook
friend
watch
NETFLIX

Map/Reduce


Input Files
Each line passed to inividual mapper instances
Map Key Value Splitting
Sort and Shuffle
Reduce Key Value Pairs
Final Output
Apple Orange Mango
Orange Grapes Plum
Apple Plum Mango
Apple Apple Plum
Apple Orange Mango
Orange Grapes Plum
Apple Plum Mango
Apple Apple Plum
Apple,1
Orange,1
Mango,1
Orange,1
Grapes,1
Plum,1
Apple,1
Plum,1
Mango,1
Apple,1
Apple,1
Plum,1
Apple,1
Apple,1
Apple,1
Apple,1
Grapes,1
Mango,1
Mango,1
Orange,1
Orange,1
Plum,1
Plum,1
Plum,1
Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3
Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3

Map/Reduce

Map

```
(
  String key,
  String value
):
// key: document name
// value: document contents
for each word w in value:
  EmitIntermediate(w, "1");
```

Reduce

```
(
  String key,
  Iterator values
):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
  result += ParseInt(v);
Emit(AsString(result));
```

Map

Reduce

The Big Secret of Big Data

- “Dirty secret of Big Data is you can not able deploy if you not have SQL expert on staff.”—DevOps Borat
- Pig, Hive, Tenzing, Impala

Resources

- *Seven Databases in Seven Weeks* by Eric Redmond and Jim R. Wilson
- *NoSQL Distilled* by Pramod Sadalage and Martin Fowler
- KVLite for Windows (single-process version of Oracle NoSQL Database)
- Cloudera VM (Sun VirtualBox)
- **nosql-database.org** (your ultimate guide to the non-relational universe)

Soul-searching For The Relational Camp

- *What* makes relational so sacred?

Simplicity

- Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation).—Codd, E. F. “A relational model of data for large shared data banks.” (1970).

- Surely, in the choice of logical data structures that a system is to support, there is one consideration of absolutely paramount importance – and that is the convenience of the majority of users. … To make formatted data bases readily accessible to users (especially casual users) who have little or no training in programming we must provide the simplest possible data structures and almost natural language. … What could be a simpler, more universally needed, and more universally understood data structure than a table? Why not permit such users to view all the data in a data base in a tabular way?—Codd, E. F. “Normalized data base structure: a brief tutorial.” (1971).

The Target Audience

- There is a large class of users who, while they are not computer specialists, would be willing to learn to interact with a computer in a reasonably high-level, non-procedural query language. Examples of such users are **accountants, engineers, architects, and urban planners**. It is for this class of users that SEQUEL is intended.—Chamberlin, Donald, and Raymond Boyce. “SEQUEL: A Structured English Query Language.” (1974).

Power

- Codd gave a seminar and a lot of us went to listen to him. This was as I say a revelation for me because Codd had a bunch of queries that were fairly complicated queries and since I'd been studying CODASYL, I could imagine how those queries would have been represented in CODASYL by programs that were five pages long that would navigate through this labyrinth of pointers and stuff. Codd would sort of write them down as one-liners. These would be queries like, "Find the employees who earn more than their managers." He just whacked them out and you could sort of read them, and they weren't complicated at all, and I said, "Wow." This was kind of a conversion experience for me, that I understood what the relational thing was about after that.—Donald Chamberlin, the creator of the SQL language in The 1995 SQL Reunion: People, Projects, and Politics

Derivability, Redundancy and Consistency of Relations

- The large, integrated data banks of the future will contain many relations of various degrees in stored form. It will not be unusual for this set of stored relations to be redundant. Two types of redundancy are defined and discussed. One type may be employed to improve accessibility of certain kinds of information which happen to be in great demand. When either type of redudnancy exists, those responsible for control of the data bank should know about it and have some means of detecting any “logical” inconsistencies in the total set of stored relations. Consistency checking might be helpful in tracking down unauthorized (and possibly fraudulent) changes in the data bank contents.—Codd, E. F. “Derivability, Redundancy and Consistency of Relations Stored in Large Data Banks.” (1969).

Assertions

- A pilot may fly a certain type of aircraft only if (1) he has flown this type of aircraft previously or (2a) he has attended an appropriate classroom training course and (2b) his instructor is one of the co-pilot.
- If this is the first time the pilot is going to be flying this type of aircraft then, prior to the flight date, one of the co-pilots must have led the classroom training course mandated for this type of aircraft and the pilot must have attended and passed the course.

Assertions

```
CREATE ASSERTION employees_a1
AS CHECK
(
  (SELECT COUNT(*) FROM employees) >= 50
)
```

Assertions

```
CREATE ASSERTION employees_departments_fk
AS CHECK
(
  NOT EXISTS
  (
    SELECT * FROM employees e
    WHERE NOT EXISTS
    (
      SELECT * FROM departments d
      WHERE d.department_id = e.department_id
    )
  )
)
```

Take-Home Message

THE RELATIONAL MODEL IS SACRED BECAUSE IT GIVES APPLICATION SOFTWARE DEVELOPERS THE ABILITY TO ASSERT AND ENFORCE CONSISTENCY OF DATA IN DATABASES.

Soul-searching For The Relational Camp

- The mistakes of the relational camp

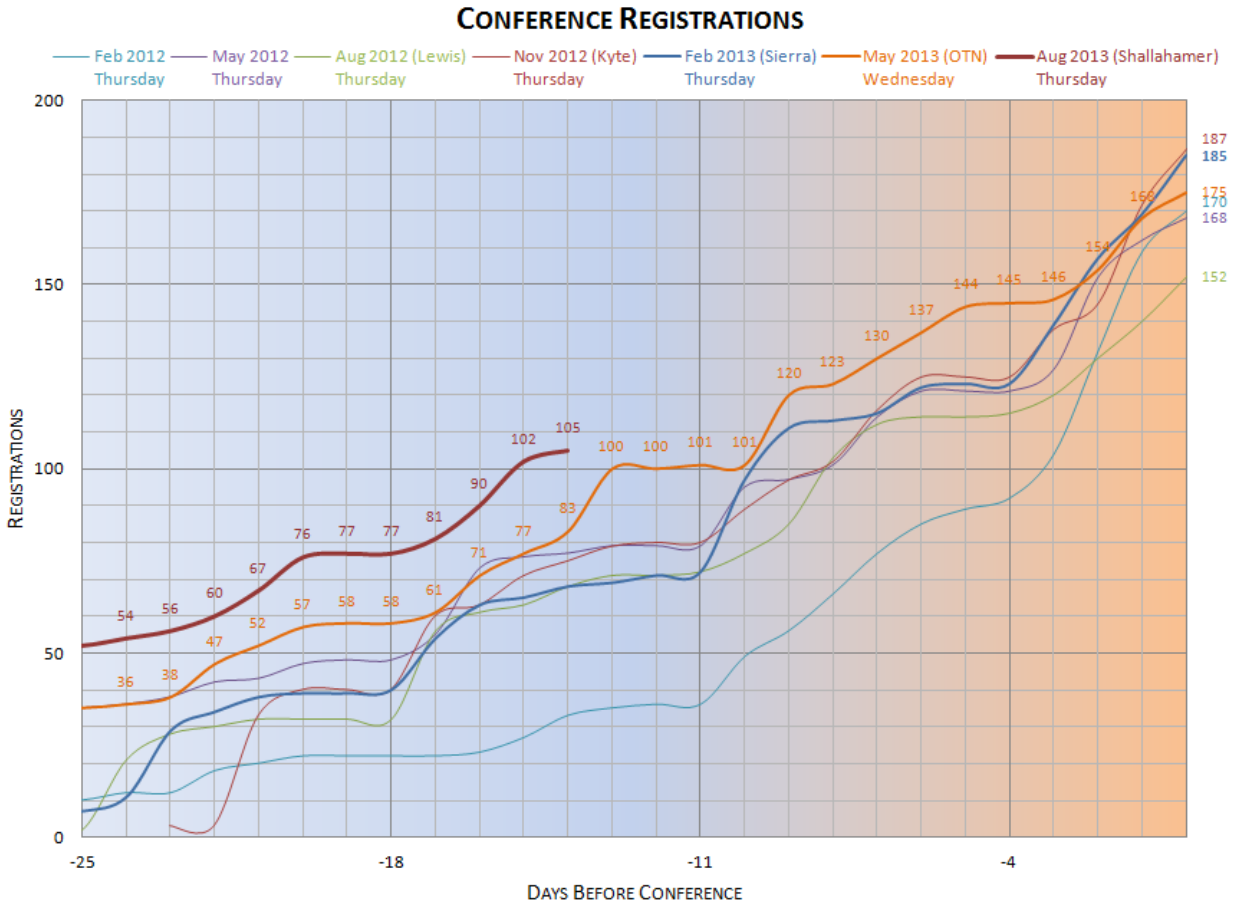
Codd's First Mistake—His Biggest Mistake

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation).—Codd, E. F. "A relational model of data for large shared data banks." (1970).

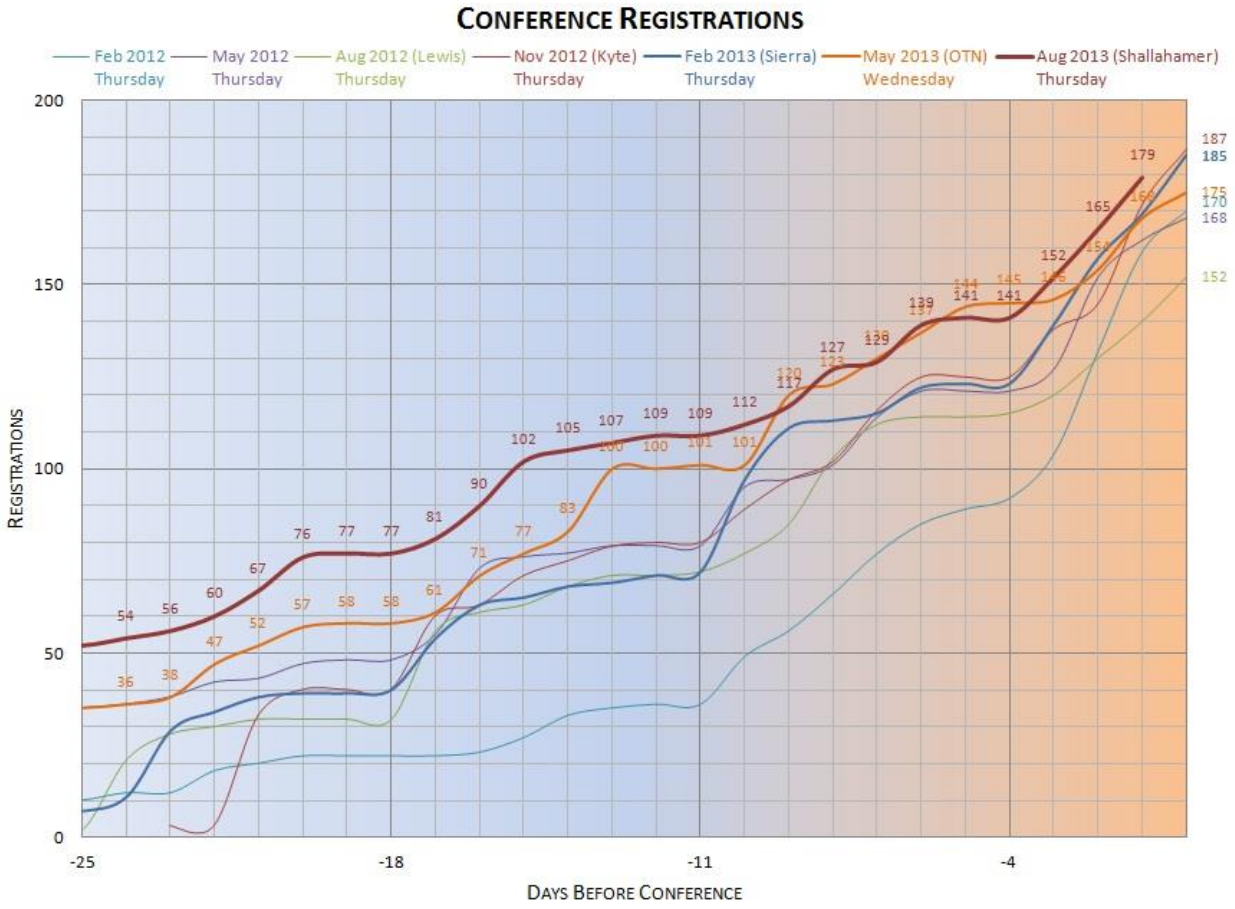
The target audience for SQL

- There is a large class of users who, while they are not computer specialists, would be willing to learn to interact with a computer in a reasonably high-level, non-procedural query language. Examples of such users are **accountants, engineers, architects, and urban planners**. It is for this class of users that SEQUEL is intended.—Chamberlin, Donald, and Raymond Boyce. “SEQUEL: A Structured English Query Language.” (1974).

Conference Registrations



Conference Registrations



Codd's Second Mistake

- **employee** (man#, name, birthdate, **jobhistory** (jobdate, title, **salaryhistory** (salarydate, salary)), **children** (childname, birthyear))

Normalized Set

- **employee** (man#, name, birthdate, jobhistory, children)
- **jobhistory** (man#, jobdate, title, salaryhistory)
- **salaryhistory** (man#, jobdate, salarydate, salary)
- **children** (man#, childname, birthyear)

Normalized set

Codd's Second Mistake—First Normal Form

- A relation whose domains are all simple can be represented in storage by a two-dimensional column-homogeneous array of the kind discussed above. Some more complicated data structure is necessary for a relation with one or more nonsimple domains.
- The simplicity of the array representation which becomes feasible when all relations are cast in normal form is not only an advantage for storage purposes but also for communication of bulk data between systems which use widely different representations of the data.
- The second-order predicate calculus (rather than first-order) is needed because the domains on which relations are defined may themselves have relations as elements.

The Problem With Flat Tables

- *"Using [flat] tables to store objects is like driving your car home and then disassembling it to put it in the garage. It can be assembled again in the morning, but one eventually asks whether this is the most efficient way to park a car."*—Dyson, Esther. *Review 1.0,* September 1988.

However, Codd never said that …

- … each stored table should occupy one physical file
- … data should be stored in row-major order
- … stored tables have only one storage representation each
- … data should be stored in normalized form only
- … a single data block only contain data from a single table
- … data should not be *stored* in compact forms

Codd's Third Mistake—Relational Calculus

- "Clearly, the majority of users should not have to learn either the relational calculus or algebra in order to interact with data bases. However, requesting data by its properties is far more natural than devising a particular algorithm or sequence of operations for its retrieval. Thus, a calculus-oriented language provides a good target language for a more user-oriented source language."

Relational Languages

```
SELECT
  first_name,
  last_name
FROM employees mgr
WHERE EXISTS (
  SELECT *
  FROM employees emp
  WHERE emp.manager_id =
mgr.employee_id
  AND emp.salary >
mgr.salary
)
```

Relational Calculus

```
SELECT DISTINCT
  mgr.first_name,
  mgr.last_name
FROM employees mgr
INNER JOIN employees
emp
ON emp.manager_id =
mgr.employee_id
WHERE emp.salary >
mgr.salary
```

Relational Algebra

Codd's Fourth Mistake—Normalization?

- "Normalization is a step-by-step reversible process of replacing a given collection of relations by successive collections in which the relations have a progressively simpler and more regular structure. The objectives of normalization are … To free the collection of relations from undesirable insertion, update and deletion dependencies" —Codd, E. F. "Normalized data base structure: a brief tutorial." (1971).

Heard on AskTom

- “I became interested in the CBO’s selectivity calculations trying to understand why it comes up with some of the ridiculously low cardinality estimates (like 1 when in reality there are 80,000+) which then lead to disastrous access plans that take hours, provided they finish at all, instead of minutes or seconds.”—Wolfgang Breitling, author of “Tuning by Cardinality Feedback”

Codd's Fifth Mistake—The Marriage of Relational with DBMS

- Codd, E. F. "Is your DBMS really relational?" *ComputerWorld,* October 14, 1985.

- Codd, E. F. "Does your DBMS run by the rules?" *ComputerWorld,* October 32, 1985.

The sixth mistake—SQL-92 Assertions not implemented

- Unavailable in Oracle Database, SQL Server, DB2, MySQL, PostgreSQL
- Limited support in Oracle Rdb
  - The predicate in a CHECK table constraint can refer directly to any column in the table and can refer to columns in other tables in the database through column select expressions in the predicate.

We Don't Use Databases—A story

"We don't use databases. We don't use indexes. We store all our data in compressed text files. Each compressed text file contains one year of data for one location. There is a separate subdirectory for each year. We have a terabyte of data going back to 1901 so we currently have 113 subdirectories. The performance is just fine, thank you."—
http://iggyfernandez.wordpress.com/2013/01/22/we-dont-use-databases-we-dont-use-indexes/

Soul-searching For The Relational Camp

- We Don't Use Databases—Demonstration
  Refer to WeDontUse12c.sql and WeDontUse12c.log
  Instructions available at http://iggyfernandez.wordpress.com/2013/01/22/we-dont-use-databases-we-dont-use-indexes/

Bonus Slide—
NewSQL from VoltDB

- Shared-nothing cluster (sharded)
- Each shard is replicated K times ("K safety")
- For maximum speed, only one transaction operates in a shard at a time (no locking, no latching, no concurrency)
- Works well only for shardable schemas
- Limited subset of SQL only
- SQL procedures compiled linked into the execution engine

Bonus Questions

- Performance, scalability, and reliability without NoSQL?
- What about that CAP theorem?
- Should I study NoSQL?
- Can I make a career transition?

Recap

- Amazon requirements: Extreme performance, scalability, and availability
- Amazon solution: Functional segmentation, Sharding, Multi-master replication, BLOBs
- eBay has the same goals as Amazon but uses Oracle and SQL for it's e-commerce platform
- Clusters

Thank You!

Q & A

iggy_fernandez@hotmail.com
http://iggyfernandez.wordpress.com