Big Data Platforms
How Big is Big?

The Data
<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Size/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1000^2$ (5.3 MB)</td>
<td>Complete works of Shakespeare (text)</td>
</tr>
<tr>
<td>$1000^3$ (~5-500 GB)</td>
<td>Your data</td>
</tr>
<tr>
<td>$1000^4$ (10 TB)</td>
<td>Library of Congress (text)</td>
</tr>
<tr>
<td>$1000^4$ (200 TB)</td>
<td>Longest machine-generated proof</td>
</tr>
<tr>
<td>$1000^5$ (~1 PB/day)</td>
<td>Data uploaded to Facebook</td>
</tr>
<tr>
<td>$1000^5$ (~1 PB/day)</td>
<td>Data generated by LHC sensors</td>
</tr>
<tr>
<td>$1000^6$ (~2.5 EB/day)</td>
<td>Data produced by humanity</td>
</tr>
<tr>
<td>&gt; 1 trillion (10^12) \times Shakespeare</td>
<td></td>
</tr>
</tbody>
</table>
How Big is Big?
The Infrastructure
100 MW

1-10 power plants

80,000 US households
Storage
Relational Databases

- Application
- Front end (SQL)
- Query processor
- Transaction processor
- File system
Relational Databases

**Strengths**
Declarative query language (SQL)
Query optimization
Transactions
   Atomicity, consistency, isolation, durability

**Challenges at scale**
Concurrency control
Reliability
Replication
BigTable

Basic idea: key-value store

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>34M</td>
</tr>
<tr>
<td>France</td>
<td>64M</td>
</tr>
<tr>
<td>Germany</td>
<td>82M</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>USA</td>
<td>307M</td>
</tr>
</tbody>
</table>

lookup(key) → value
scan(key range) → values
insert(key, value)
delete(key)
### BigTable

#### Tablet server 1

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k₁</td>
<td>v₁</td>
</tr>
<tr>
<td>k₂</td>
<td>v₂</td>
</tr>
<tr>
<td>k₃</td>
<td>v₃</td>
</tr>
<tr>
<td>k₄</td>
<td>v₄</td>
</tr>
<tr>
<td>k₅</td>
<td>v₅</td>
</tr>
<tr>
<td>k₆</td>
<td>v₆</td>
</tr>
<tr>
<td>k₇</td>
<td>v₇</td>
</tr>
<tr>
<td>k₈</td>
<td>v₈</td>
</tr>
<tr>
<td>k₉</td>
<td>v₉</td>
</tr>
</tbody>
</table>

#### Tablet server 2

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k₁</td>
<td>v₁</td>
</tr>
<tr>
<td>k₂</td>
<td>v₂</td>
</tr>
<tr>
<td>k₃</td>
<td>v₃</td>
</tr>
<tr>
<td>k₄</td>
<td>v₄</td>
</tr>
</tbody>
</table>

#### Tablet server 3

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k₅</td>
<td>v₅</td>
</tr>
<tr>
<td>k₆</td>
<td>v₆</td>
</tr>
</tbody>
</table>

---

**CS 102**

**Big Data Platforms**
Spanner

= BigTable
  + transactions
  + global replication
Comparison

**SQL**  Relational databases
Relations, queries, optimization, transactions
Costly to scale

**NoSQL**  BigTable
Scale
No data structure, queries, optimization, (multi-row) transactions
Eventual consistency

**NewSQL**  Spanner
Scale
Transactions
Processing
MapReduce

Original application: building a text index

Loading

Tokenizing

Sorting

Disk

Intermediate runs

Flushing

Page stream

ant
dog

1

( ant, 1 )
( dog, 1 )
( dog, 2 )
( ant, 3 )
( dog, 3 )

2
dog
cat

( ant, 1 )
( dog, 1 )
( cat, 2 )
( dog, 1 )
( dog, 2 )
( dog, 3 )

3

ant
dog

( ant, 1 )
( ant, 3 )
( cat, 2 )
( dog, 1 )
( dog, 2 )
( dog, 3 )
MapReduce

Intermediate runs

( ant, 1 )
( ant, 3 )
( cat, 2 )
( dog, 1 )
( dog, 2 )
( dog, 3 )

( ant, 5 )
( cat, 4 )
( dog, 4 )
( dog, 5 )
( eel, 6 )

Merge

( ant, 1 )
( ant, 3 )
( ant, 5 )
( cat, 2 )
( cat, 4 )
( dog, 1 )
( dog, 2 )
( dog, 3 )
( dog, 4 )
( dog, 5 )
( eel, 6 )

Final index

( ant: 1, 3, 5 )
( cat: 2, 4 )
( dog: 1, 2, 3, 4, 5 )
( eel: 6 )
MapReduce

Loading

Tokenizing

Sorting

Disk

Intermediate runs

Flushing

Page stream

Map

ant
dog

1

ant
dog

2
dog
cat

3

ant
dog

( ant, 1 )
( dog, 1 )
( dog, 2 )
( cat, 2 )
( ant, 3 )
( dog, 3 )
MapReduce

Intermediate runs

( ant, 1 )
( ant, 3 )
( cat, 2 )
( dog, 1 )
( dog, 2 )
( dog, 3 )

( ant, 5 )
( cat, 4 )
( dog, 4 )
( dog, 5 )
( eel, 6 )

Merge

( ant, 1 )
( ant, 3 )
( ant, 5 )
( cat, 2 )
( cat, 4 )
( dog, 1 )
( dog, 2 )
( dog, 3 )
( dog, 4 )
( dog, 5 )
( eel, 6 )

Reduce

( ant, 1 )
( ant, 3 )
( ant, 5 )
( cat, 2 )
( cat, 4 )
( dog, 1 )
( dog, 2 )
( dog, 3 )
( dog, 4 )
( dog, 5 )
( eel, 6 )

( ant: 1, 3, 5 )
( cat: 2, 4 )
( dog: 1, 2, 3, 4, 5 )
( eel: 6 )

Final index
MapReduce

Framework
MapReduce

Example: counting word occurrences

map(String docId, String docBody):
  for each word w in docBody
    emitIntermediate(w, 1)

E.g., map(42, “cat dog cat bat dog”) emits
[“cat”, 1], [“dog”, 1], [“cat”, 1], [“bat”, 1], [“dog”, 1]
MapReduce

Example: counting word occurrences

reduce(String key, Iterator values):
    int result = 0
    for each v in values result += v
    emit(result)

E.g., reduce(“dog”, { 1, 1, 1, 1 }) emits 4
Spark

Generalized data flow
Resilient distributed data sets (RDDs)
Transformations
Actions
Platforms
Cloud Computing

“The fight to dominate cloud computing will increase competition and innovation”
(*Battle of the clouds*, October 2009)

“Tech giants are waging a price war to win other firms’ computing business”
(*Silver lining*, August 2014)

“As cloud-computing prices keep falling, the whole IT business will change”
(*The cheap, convenient cloud*, April 2015)
Cloud Computing

What is it?
On-demand access to shared computing resources
Infrastructure/platform/software as a service
Hardware virtualization
Business model (focus on business, not infra, differentiators)
Cloud Platforms

Amazon Web Services
Google Cloud
Microsoft Azure

Application | Big data
-------------|-------------
Compute      | Storage
Virtual machine | File system
Network
Compute and Application Engines

Virtual machines
1-32 CPUs, 1-200 GB RAM
0-3 TB local HDD or SSD
0-64 TB network storage
Linux or Windows

Networking
Virtual networks
Load balancing
Domain name system
Storage

Basic storage
Distributed files

Relational database
E.g., MySQL instances (16 cores, 100 GB RAM, read replicas)

Distributed data store
E.g., BigTable (hundreds of PBs, millions of ops/second)
Big Data

Procedural and semi-declarative processing
MapReduce/Hadoop, Spark, Pig

Declarative queries
SQL-like query language
Integrations with, e.g., Tableau visualization

Interactive processing
E.g., on top of Jupyter (Python, SQL, JavaScript)
Other Services

**Machine learning**
Libraries, e.g., TensorFlow
Vision, speech recognition, translation APIs

**Tools**
Management
Development
Identity and security
Summary

Big data and infrastructure
Storage
Processing
Cloud platforms

Questions?