

**Google** Cloud Platform

# Introducing Google BigQuery

BigQuery for Data Analysts

V1.2

*Approximate timing: 45 minutes*

# Agenda

- 1 Current State of Big Data
- 2 Big Data *the Cloud Way*
- 3 What is BigQuery?
- 4 BigQuery Use Cases and Customer Stories
- 5 Demo, Quiz & Lab

# Big Data Current State (1 of 2)

Big Data growth is **explosive**, and accumulating **exponentially**

## *Varied sources and types*

- Real-Time Streaming Data
- Structured and Unstructured Data
- Web, Social, Sensor
- Audio, Video, Documents



## *Countless uses*

- Discover New Trends and Insights
- Explore New Business Opportunities
- Increase Innovation
- Fraud Detection
- Prediction Models
- Identify Inefficiencies
- Augment decisions with quantitative statistics

## Notes:

"There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days." – Eric Schmidt, of Google said in 2010

"Information is the oil of the 21st century, and analytics is the combustion engine." – Peter Sondergaard, Gartner Research

## Big Data Current State (2 of 2)

- Big Data amplifies resource shortages
  - Big Data analytics is constantly evolving and demand continues to build
  - Most development methods do not accommodate data discovery and interactive analytics
  - Large scale on-premise computing resources are capital and set-up time intensive
  - Capitalizing on Big Data insights requires dynamic system resizing and reconfiguration

### Notes:

Agile methods are gaining acceptance in big data analytics. A discovery/model phase needs to be done as a precursor to the traditional (including agile) development phase.

#### **With Big Data, Data discovery helps define applications Analytics targets.**

Replace traditional requirements specification processes, which are fundamentally flawed because we don't know "requirements" until we actually interact with the production data.

The "v1" versions of the business discovery "application" discovery can be used in production, in parallel with spinning off the requirements earmarked for hardening.

When the "hardened" version of the requirements are ready for production, the business discovery visualizations are "re-honed" to be fed from the hardened asset.

### Discover/Model:

- Experiment/ Refine (exploration; collaboration)

- Support (Super Expert; Governance)

- Decide (Episodic; "red herring"; Productize)

### Expand (Development)

- Add Data (New objects; integrations; Data Release Mgmt; Data Chg Mgmt)

Add Capability(Dev methods, tools, Release Mgmt; Change Mgmt)

Extend Base (Engagement methods)

Steady State (Production)

Monitor (SLA; Data Quality mgmt; Capacity mgmt; performance monitoring)

Maintain (defect mgmt; infrastructure mgmt)

Support (customer care; tech support)

# Agenda

- 1 Current State of Big Data
- 2 Big Data *the Cloud Way*
- 3 What is BigQuery?
- 4 BigQuery Use Cases and Customer Stories
- 5 Demo, Quiz & Lab

# On Premises Versus Cloud

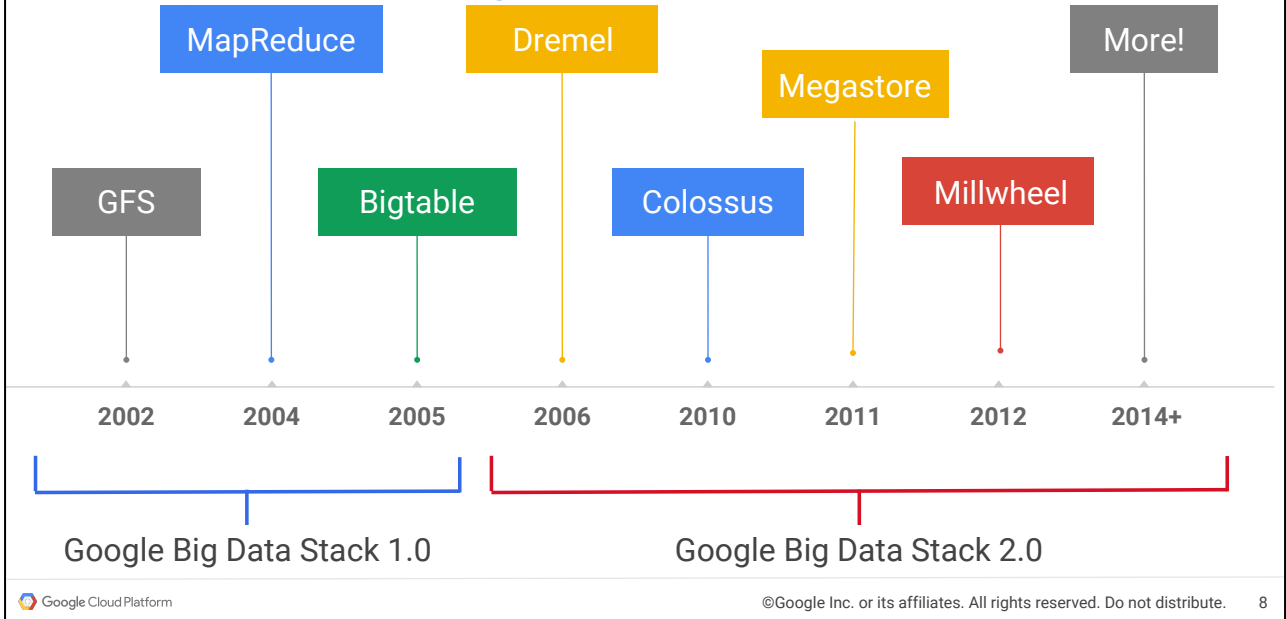
On Premises	Cloud
Exponential costs: 2x processing power equals 5x the cost	<b>Managed service:</b> Infrastructure deployed, managed, and upgraded to make it scalable and reliable
Single points of failure	<b>Cost-effectiveness:</b> Removal of operations work; Auto-scale and optimize your infrastructure consumption
Limited protection against software/hardware failure	<b>Safe and easy collaboration:</b> One version of the data and authorized users can access it without affecting job performance
Requires managed infrastructure support	

## Notes:

NoOps – The cloud provider worries about the infrastructure.

Cost Effectiveness - In addition to increased ease of use and agility, a “NoOps” solution provides clear cost benefits via the removal of operations work; but the cost benefits of big data the cloud way go even further – the platform auto-scales and optimizes your infrastructure consumption, and eliminates unused resources like idle clusters.

# Google's History of Innovation in Big Data



## Notes:

### Big Data Stack 1.0

In the early 2000's Google laid the foundation for Big Data Strategy –

- Design software with failure in mind
- Use only commodity components
- The cost of twice the amount of capacity should not be considerably more than the cost of twice the amount of hardware
- Be consistent

These principles inspired new computation architectures

- GFS – a distributed, cluster-based filesystem, GFS assumes that any disk can fail so data is stored in multiple locations
- MapReduce – A computing paradigm that divides problems into parallelized pieces across a cluster of machines
- Bigtable – Enables structured storage to scale out to multiple servers

### Big Data Stack 2.0

2.0 refined the ideas from the 1.0 stack

- Dremel – A distributed SQL query engine that can perform complex queries over data stored on GFS, Colossus, and others – the basis of BigQuery



- Colossus – A distributed file system that resolves some of the limitations with GFS
- Megastore – A geographically replicated, consistent NoSQL-type data store that insures consistent reads and writes
- Millwheel – Fault-tolerant stream processing at internet scale

# Agenda

- 1 Current State of Big Data
- 2 Big Data *the Cloud Way*
- 3 What is BigQuery?
- 4 BigQuery Use Cases and Customer Stories
- 5 Demo, Quiz & Lab

# What is BigQuery (1 of 3)

- **Fully-managed, analytics data warehouse**
  - Provides **near real-time interactive analysis** of massive datasets
  - Runs on Google's fully managed, secure, high-performance infrastructure
  - "NoOps" - No administration for performance and scale
- **Reliable**
  - Data replicated across multiple data centers
- **Economical**
  - Only pay for storage and processing used

## Notes:

We are seeing a shift in the analytics landscape. Hosted services with a simple model is empowering the data analysts to start their data analytics project much quicker. Instead of having to request for hardware/software resources from IT, justifying the upfront CAPEX for procuring hardware/software, data analysts can quickly upload their data to BigQuery. Use the tools that they are comfortable with, notably SQL or third party analytic tools, and get experiment with hypothesis very quickly. For details, see:

<https://cloud.google.com/developers/articles/getting-started-with-google-big-query>

- BigQuery is a highly parallel architecture that can process large volumes of data quickly
- BigQuery helps to eliminate customer on-premises infrastructure costs
- Data is replicated across multiple disks in multiple Google data centers
- Highly secure using Google's multi-tier security

Google infrastructure is virtually limitless in terms of storage, and processing power. There are no storage limits on data. Processing is both distributed and parallelized across Google's data centers.

Google provides a simple and pricing model. Pricing is based on the amount

of data stored in datasets and along with the amount of data processed per query. Google does not charged for the first 1TB of data processed per month.

# What is BigQuery (2 of 3)

- **Secure**

- Secured through Access Control Lists (ACLs) and Identity and Access Management (IAM)
- Data is encrypted in transport and at rest

- **Auditable**

- Google Cloud Audit Logs track Admin Activity and Data Access
- Immutable logs - “who did what, where, and when?” in BigQuery

- **Scalable**

- Virtually unlimited data storage and processing power  
Highly parallel/distributed process model

# What is BigQuery (3 of 3)

- **Flexible**

- Streaming ingestion: 100K rows/sec per table for real-time data
- Data mashup: JOIN across diverse datasets/projects

- **Easy to use**

- Data stored in denormalized **tables** (simple schemas)
- Columnar storage for high performance
- Requires no indexes, keys, or partitions
- Familiar SQL interface and intuitive UI
- Nested and repeated field support for schema flexibility
- Supports open standards - Analysts can use preferred tools

## Notes:

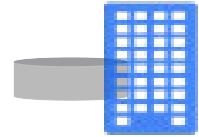
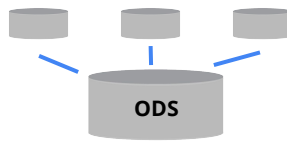
No Complex data structure required

- Simple denormalized data structure
- Data can be loaded in CSV or JSON format

Supports open standards

- SQL-like language
- Standard BI/ETL tools supported
- REST API support for integrating analytics programmatically

# BigQuery Is Not...



Transactional RDBMS	Operational Data Store	On-premises solution or appliance
BigQuery is not an OLTP system	BigQuery is not geared towards capturing live data and applying updates/deletes as they happen in the system of record	BigQuery is a self-contained, cloud-based solution

## Notes:

### Transactional Relational Database Management System (RDBMS)

- BigQuery is not an OLTP system
- BigQuery is designed for large scale, high performance OLAP workloads
- Data is stored in BigQuery in Columnar fashion
- BigQuery tables are immutable, but...BigQuery allows for fast materialization, its operations are atomic, and metadata operations are incredibly fast. Thus, updates/deletes batched together are very very easy to put together

### Operational Data Store

- BigQuery is not an ODS to apply updates/deletes as they happen in SOR
- BigQuery may be used for data warehouses, instead
- BigQuery is specifically designed for large data sets
- BigQuery executes queries requiring massive processing

### On-premise solution or Appliance

- BigQuery is available only in Google Cloud in a fully hosted

- fashion
- “NoOps”: No administration for performance and scale



# Comparisons (1 of 2)

- **OLTP** (Online Transaction Processing)
  - Handles row-level operations better (strong consistency, ACID transactions) but requires indexes and does not scale
  - BigQuery handles analytics better - Eventual consistency, no indexes, massive scale
- **OLAP** (Online Analytical Processing)
  - Similar in use cases they support
  - BigQuery allows querying via SQL

## Notes:

OLAP: BigQuery's nested and repeated fields can approximate the data cubes in OLAP but allow querying via SQL rather than MDX. MDX stands for MultiDimensional eXpressions language used for querying and manipulating multidimensional data stored in OLAP cubes.

## Comparisons (2 of 2)

- **MapReduce**

- Fundamentally a batch oriented technology
- Higher latency than BigQuery (which is near real-time)

- **NoSQL**

- Less scalable than BigQuery
- Awkward or impossible to query - No query language

### Notes:

Both BigQuery and MapReduce (Hadoop) targets big data analysis. However, they are based on different technologies. MapReduce leverages distributed file system, and co-locate compute with data, so as to process large volume of data in parallel. BigQuery leverages columnar data format, and multi-level execution trees, it is able to run aggregation queries over millions of rows in seconds.

Map (Shuffle) Reduce serves a different purpose:

- Batch based
- Shuffle is slow
- May need to do multiple queries
- Primarily to apply computational logic to the data
- Can read/write operations

BigQuery provides a way to perform interactive ad-hoc queries. It's an OLAP that complements an overall data management solution. You don't want to spend hours indexing something just to see if a query is interesting.

# Agenda

- 1 Current State of Big Data
- 2 Big Data *the Cloud Way*
- 3 What is BigQuery?
- 4 BigQuery Use Cases and Customer Stories
- 5 Demo, Quiz & Lab

# BigQuery Use Cases

<b><i>Games and social media analytics</i></b>	Near real-time insights into the behavior of game players and app users
<b><i>Advertising campaign optimization</i></b>	Analyze huge data sets to target ads more effectively
<b><i>Sensor data analysis</i></b>	Stream sensor data to allow for near real-time analysis
<b><i>POS-Retail Analytics</i></b>	Analyze sales patterns and shopper behavior
<b><i>Web logs, machine logs, infrastructure monitoring</i></b>	Analyze advertising logs, diagnose system logs

## Notes:

**Games and social media analytics** -- Claritics, a company based in Mountain View, Ca uses BigQuery to help social and mobile game developers, advertisers and media companies gain real-time insights into the behavior of game players and app users. These insights help shape key game and app design, optimize marketing effectiveness and increase application revenue. Read full use case at <https://cloud.google.com/customers/claritics/>

Backflip Studios uses BigQuery to analyze what players are doing on their games. With these insights, we can figure out areas where players are struggling, that need improvement -- what new features and content to offer, how to better retain players, etc. -- to keep them coming back. Read full use case at <http://googlecloudplatform.blogspot.com/2014/04/backflip-studios-scales-mobile-games-with-google-cloud-platform.html>

**Advertising campaign optimization** -- Slightly, a video production company uses BigQuery to make Google AdWord campaigns more effective and productive. "We're able to see and act on trends to help us target the right

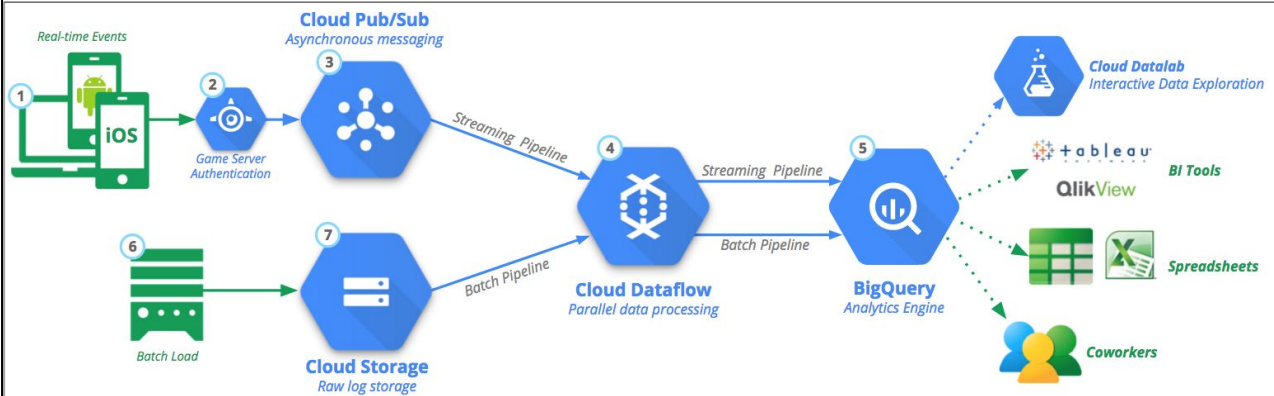
consumer with the right video at the right time. This has helped increase engagement with our ads by 300%. We are delivering more value to our customers, and growing faster and more profitably as a result.” —John Zdanowski, co-founder and chief financial officer, Sightly. Read full use case at <https://cloud.google.com/customers/sightly/>

**Sensor Data analysis** -- The internet of things (IOT) is a growth industry with the potential for 50 billion connected devices by 2020. BigQuery allows you to stream this sensor data to allow for near real time analysis. Refer to the Web logs use case to read about how Shine technologies is using streaming inserts to load near real time data from Google’s DoubleClick for Publishers product.

**POS-Retail Analytics** -- Interactions Marketing uses BigQuery for performing high- level data analytics, such as sales patterns and shopper behavior, that might help retailer and manufacturing clients plan ahead. After considering various solutions, the company focused on Google BigQuery. With previous exposure to transactional and loyalty-card data, Interactions was able to use BigQuery in a study that provided new insights into consumers’ behavior during snowstorms by combining point-of-sale (POS) and government meteorological data. Read full use case at <https://cloud.google.com/customers/interactions-marketing/>

**Web Logs, Machine logs, Infrastructure monitoring** -- Shine technologies analyses over 30 TB of log data. When one Shine’s biggest clients, a national telecommunications provider in Australia, needed to analyze a large amount of their business data in real time, they chose [Google’s DoubleClick](#) for Publishers product. They realized that they could configure DoubleClick to store the data in Google Cloud Storage, and then point Google BigQuery to those files for analysis, with just a couple of clicks. Read full use case at <http://googlecloudplatform.blogspot.com/2015/01/shine-technologies-reels-in-big-data.html>.

# Example Architecture: Mobile Gaming Analytics



## Build a mobile gaming analytics platform - a reference architecture

### Notes:

Popular mobile games can attract millions of players and generate terabytes of game-related data in a short burst of time. This places extraordinary pressure on the infrastructure powering these games and requires scalable data analytics services to provide timely, actionable insights in a cost-effective way.

To address these needs, a growing number of successful gaming companies use Google's web-scale analytics services to create personalized experiences for their players. They use telemetry and smart instrumentation to gain insight into how players engage with the game and to answer questions like: At what game level are players stuck? What virtual goods did they buy? And what's the best way to tailor the game to appeal to both casual and hardcore players?

A new reference architecture describes how you can collect, archive and analyze vast amounts of gaming telemetry data using Google Cloud Platform's data analytics products. The architecture demonstrates two patterns for analyzing mobile game events:

- **Batch processing:** This pattern helps you process game logs and other large files in a fast, parallelized manner. For example, leading mobile gaming company DeNA moved to BigQuery from Hadoop to get faster

- query responses for their log file analytics pipeline.
- **Real-time processing:** Use this pattern when you want to understand what's happening in the game right now. Cloud Pub/Sub and Cloud Dataflow provide a fully managed way to perform a number of data-processing tasks like data cleansing and fraud detection in real-time. For example, you can highlight a player with maximum hit-points outside the valid range. Real-time processing is also a great way to continuously update dashboards of key game metrics, like how many active users are currently logged in or which in-game items are most popular.

Some Cloud Dataflow features are especially useful in a mobile context since messages may be delayed from the source due to mobile Internet connection issues or batteries running out. Cloud Dataflow's built-in session windowing functionality and triggers aggregate events based on the actual time they occurred (event time) as opposed to the time they're processed so that you can still group events together by user session even if there's a delay from the source.

But why choose between one or the other pattern? A key benefit of this architecture is that you can write your data pipeline processing once and execute it in either batch or streaming mode without modifying your codebase. So if you start processing your logs in batch mode, you can easily move to real-time processing in the future. This is an advantage of the high-level Cloud Dataflow model that was released as open source by Google.

Cloud Dataflow loads the processed data into one or more BigQuery tables. BigQuery is built for very large scale, and allows you to run aggregation queries against petabyte-scale datasets with fast response times. This is great for interactive analysis and data exploration, like the example screenshot above, where a simple BigQuery SQL query dynamically creates a Daily Active Users (DAU) graph using Google Cloud Datalab.

## Sample Use Case - Zulily (1 of 2)

- Fast-growing e-commerce company launching more than 9,000 product styles and 100 daily sales events with 4.9 million customers
- *Challenge:*
  - Design a data platform allowing hundreds of users to make lightning fast decisions
  - Handle exponential data growth



## Sample Use Case - Zulily (2 of 2)

- *Solution:*
  - Hadoop on Google Compute Engine, data storage with Google Cloud Storage, and analytics using Google BigQuery
  - Zulily was able to roll out the solution to hundreds of users in only six months
- Read the story at:  
<https://cloud.google.com/customers/zulily/>

# Agenda

- 1 Current State of Big Data
- 2 Big Data *the Cloud Way*
- 3 What is BigQuery?
- 4 BigQuery Use Cases and Customer Stories
- 5 Demo, Quiz & Lab

# Demo

```
SELECT
    language, SUM(views) as views
FROM
    [bigquery-samples:wikipedia_benchmark.Wiki10B]
WHERE
    regexp_match(title,"G.*o.*o.*g")
GROUP by language
ORDER by views DESC
```

Sample query - Processes over 10 billion rows in less than 10 seconds

## Notes:

This query would kill a relational database. Note the cost of this query.

Table size - 693 GB

Number of rows 10,677,046,566 -- over 10 billion rows

Processing time < 10 seconds

## Module Review

Which of the following statements are true?  
(select **2** of the available options)

- ☐ Elimination of administrative costs is a BigQuery advantage
- ☐ BigQuery is a relational database
- ☐ BigQuery is proprietary
- ☐ Unlike BigQuery, MapReduce is a batch-oriented technology and is not designed for near real-time results

# Lab

Sign up for the free trial and create a project

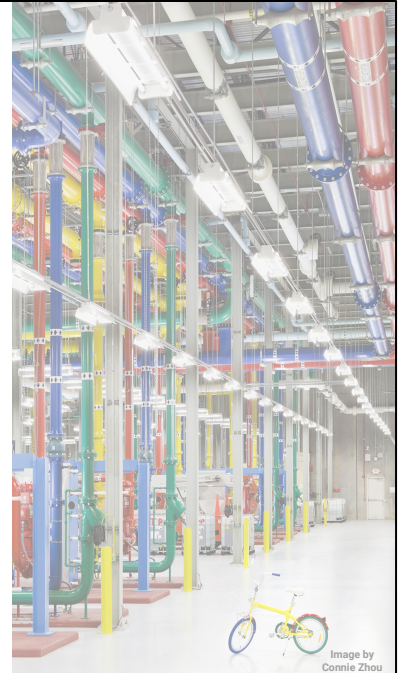


Image by  
Connie Zhou

# Resources

- What is BigQuery?  
<https://cloud.google.com/bigquery/what-is-bigquery>
- DevBytes - What is BigQuery?  
<https://www.youtube.com/watch?v=aupC-Wj7XDY>
- Dremel: Interactive Analysis of Web-Scale Datasets  
<https://research.google.com/pubs/pub36632.html>
- Customer case studies  
<https://cloud.google.com/customers/>

## Module Review Answers

Which of the following statements are true?  
(select **2** of the available options)

- ✓ Elimination of administrative costs is a BigQuery advantage
- ☐ BigQuery is a relational database
- ☐ BigQuery is proprietary
- ✓ Unlike BigQuery, MapReduce is a batch-oriented technology and is not designed for near real-time results

