Google Cloud Data Platform & Services

Gregor Hohpe

All About Data

We Have More of It

- Internet – data more easily available

- Logs – user & system behavior

- Cheap Storage – keep more of it

Beyond just Relational

Structure

- Relational (Hosted SQL)
- Record-oriented (Bigtable)
- Nested (Protocol Buffers)
- Graphs (Pregel)

Beyond just Relational

Analysis needs

- Number crunching (MapReduce / FlumeJava)
- Ad hoc (Dremel)
- Precise vs. Estimate (Sawzall)
- Model Generation & Prediction (Prediction API)

Google Storage for Developers

Store your data in Google's cloud

What Is Google Storage?

- Store your data in Google's cloud
  - any format, any amount, any time

- You control access to your data
  - private, shared, or public

- Access via Google APIs or 3rd party tools/libraries

Sample Use Cases

Static content hosting
e.g. static html, images, music, video

Backup and recovery
e.g. personal data, business records

Sharing
e.g. share data with your customers

Data storage for applications
e.g. used as storage backend for Android, App Engine, Cloud based apps

Storage for Computation
e.g. BigQuery, Prediction API

Core Features

- RESTful API
  - `GET`, `PUT`, `POST`, `HEAD`, `DELETE`
  - Resources identified by URI
  - Compatible with S3
- Organized in Buckets
  - Flat containers, i.e. no bucket hierarchy
- Buckets Contain Objects
  - Any type
  - Size: <100 GB / object
- Access Control for Google Accounts
  - For individuals and groups
- Authenticate via
  - Signing request using access keys
  - Web browser login

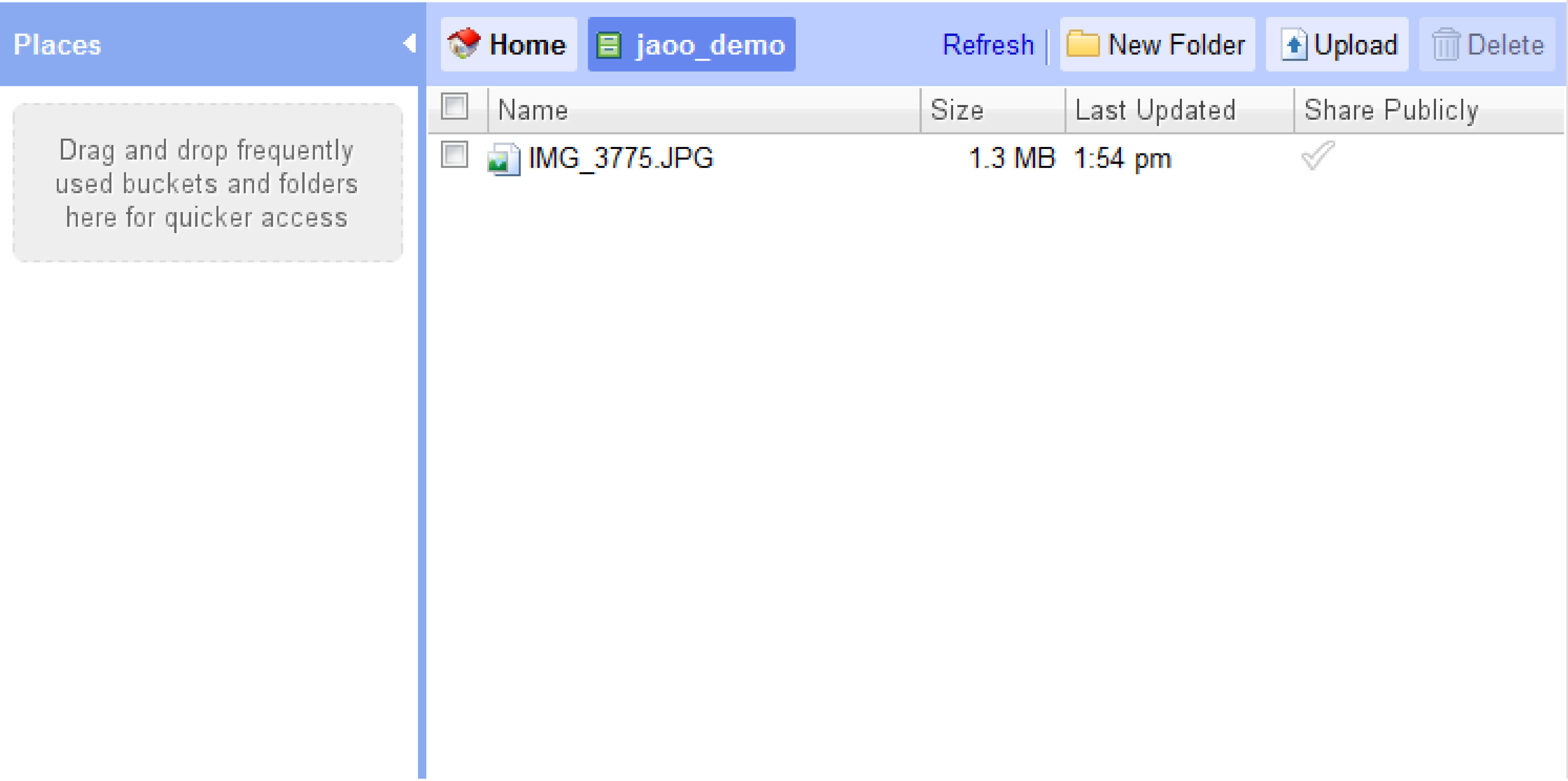
Performance and Scalability

- Objects of any type and 100 GB / Object
- Unlimited numbers of objects, 1000s of buckets

- All data replicated to multiple US data centers
- Leveraging Google's worldwide network for data delivery

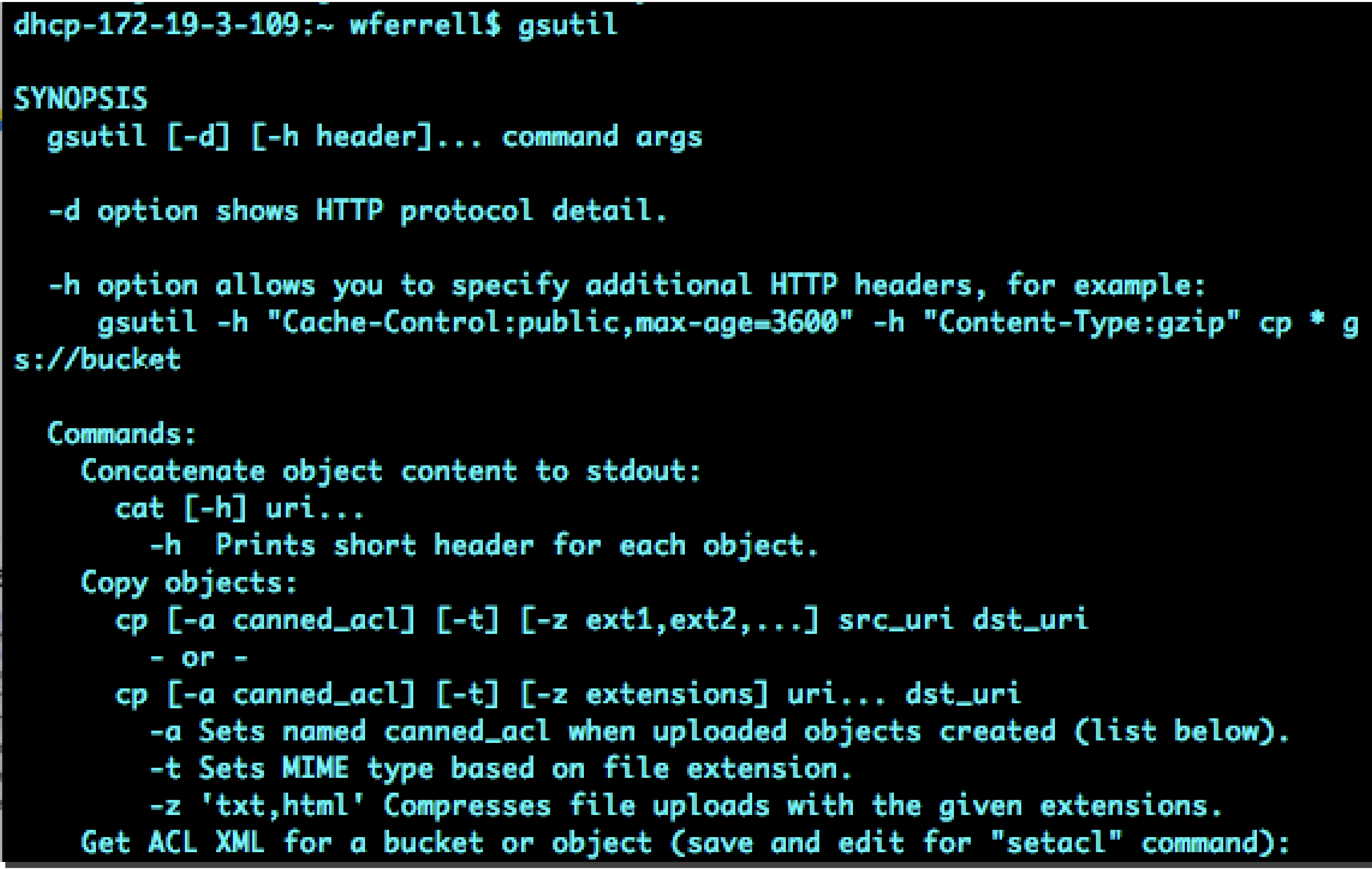
- Only you can use bucket names with your domain names
- “Read-your-writes” data consistency
- Range Get

Google Storage for Developers



Drag and drop frequently used buckets and folders here for quicker access

□	Name	Size	Last Updated	Share Publicly
□	IMG_3775.JPG	1.3 MB	1:54 pm	✓



```
dhcp-172-19-3-109:~ wferrell$ gsutil

SYNOPSIS
  gsutil [-d] [-h header]... command args

  -d option shows HTTP protocol detail.

  -h option allows you to specify additional HTTP headers, for example:
     gsutil -h "Cache-Control:public,max-age=3600" -h "Content-Type:gzip" cp * g
s://bucket

  Commands:
    Concatenate object content to stdout:
      cat [-h] uri...
        -h  Prints short header for each object.
    Copy objects:
      cp [-a canned_acl] [-t] [-z ext1,ext2,...] src_uri dst_uri
        - or -
      cp [-a canned_acl] [-t] [-z extensions] uri... dst_uri
        -a Sets named canned_acl when uploaded objects created (list below).
        -t Sets MIME type based on file extension.
        -z 'txt,html' Compresses file uploads with the given extensions.
    Get ACL XML for a bucket or object (save and edit for "setacl" command):
```

Pricing & Availability

- Storage - $0.17/GB/month
- Network
  - Upload data to Google
    - $0.10/GB
  - Download data from Google
    - $0.15/GB for Americas and EMEA
    - $0.30/GB for APAC
- Requests
  - PUT, POST, LIST - $0.01 per 1,000 Requests
  - GET, HEAD - $0.01 per 10,000 Requests

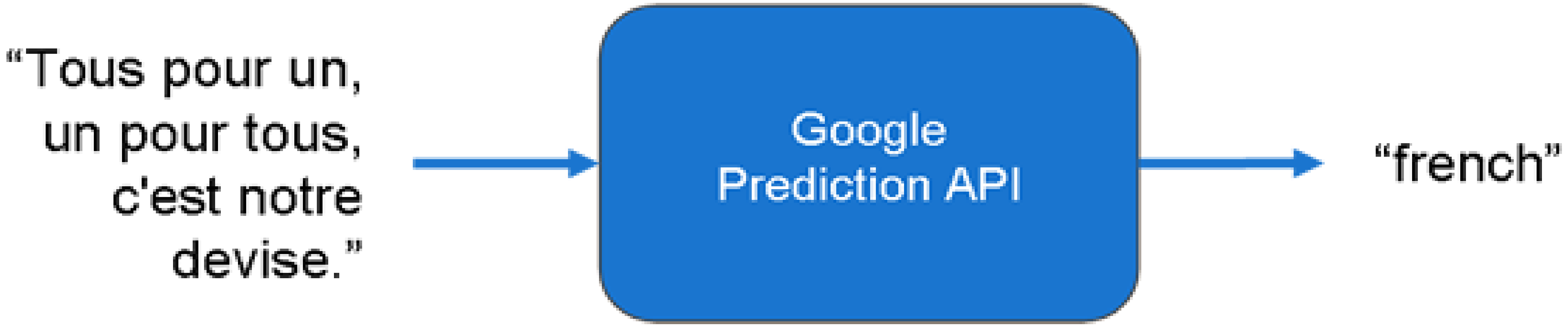
- Free storage (up to 100GB) during limited preview period
- No SLA
- http://code.google.com/apis/storage

GOOGLE PREDICTION API

GOOGLE'S PREDICTION ENGINE IN THE CLOUD

Introducing the Google Prediction API

- Google's machine learning technology
- Now available externally
- RESTful Web service



“Tous pour un,
un pour tous,
c'est notre
devise.”

Google
Prediction API

“french”

How does it work?

- 1. TRAIN
- The Prediction API finds relevant features in the sample data during training.

"english"	The quick brown fox jumped over the lazy dog.
"english"	To err is human, but to really foul things up you need a computer.
"spanish"	No hay mal que por bien no venga.
"spanish"	La tercera es la vencida.

2. PREDICT
The Prediction API later searches for those features during prediction.

?	To be or not to be, that is the question.
?	La fe mueve montañas.

Countless applications...

- Customer
- Sentiment

Transaction Risk

Species Identification

Message Routing

Diagnostics

Churn Prediction

Legal Docket Classification

Suspicious Activity

Work Roster Assignment

Inappropriate Content

Recommend Products

Political Bias

Uplift Marketing

Email Filtering

Career Counseling

... and many more ...

Using Your Data with Prediction API & BigQuery

1. Upload

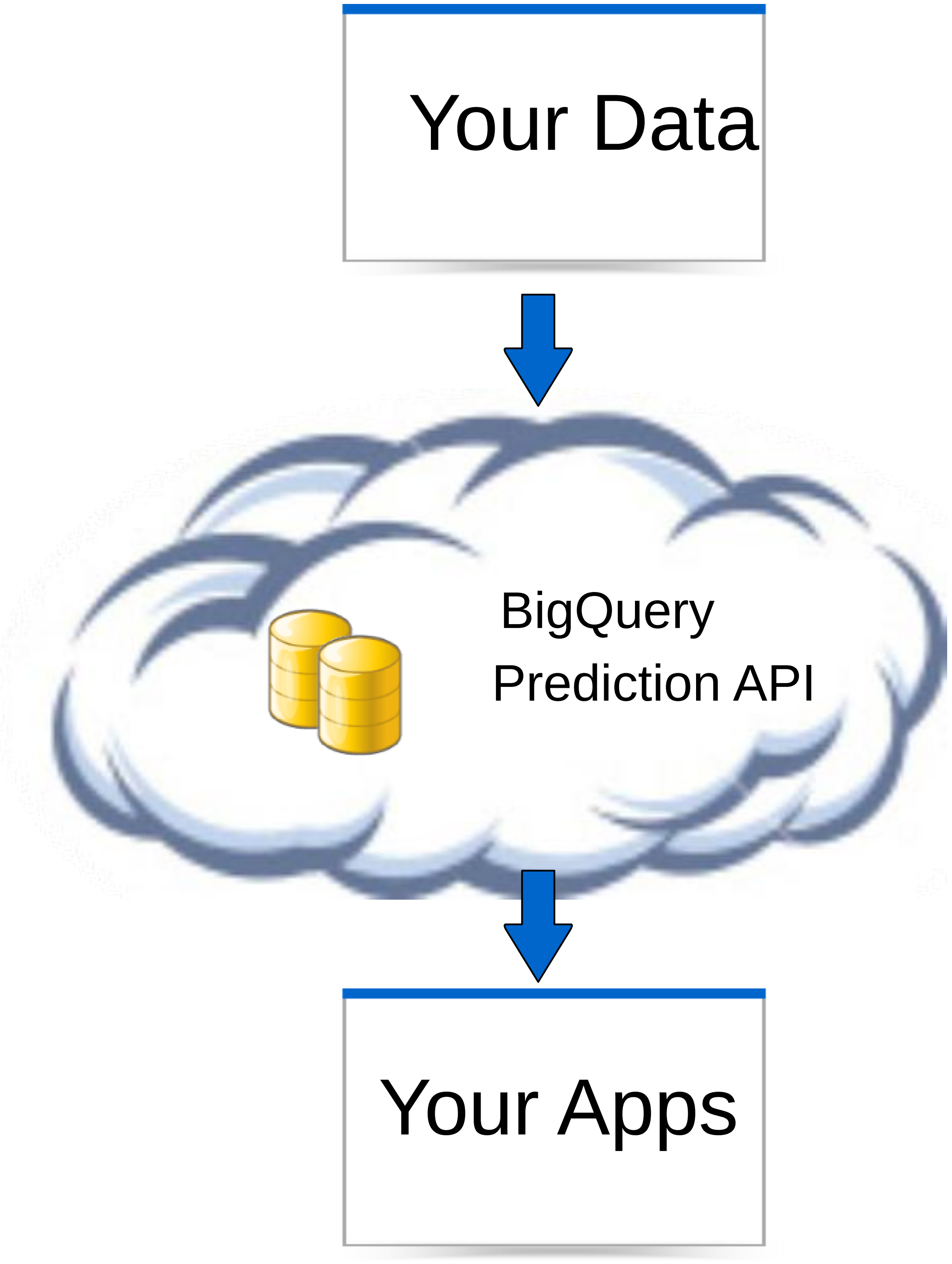
Upload your data to Google Storage

2. Process

Import to tables
Train a model

3. Act

Run queries
Make predictions



Step 1: Upload

- Training data: outputs and input features
- Data format: comma separated value format (CSV), result in first column

```
"english","To err is human, but to really ..."
"spanish","No hay mal que por bien no venga."
...
```

Upload to Google Storage

```
gsutil cp ${data} gs://yourbucket/${data}
```

Step 2: Train

- To train a model:

  POST prediction/v1.1/training?data=mybucket%2Fmydata

- Training runs asynchronously. To see if it has finished:

  GET prediction/v1.1/training/mybucket%2Fmydata

```
{"data":{
   "data":"mybucket/mydata",
   "modelinfo":"estimated accuracy: 0.xx"}}}
```

Step 3: Predict

```
POST prediction/v1.1/query/mybucket%2Fmydata/predict

{ "data":{
    "input": { "text" : [
      "J'aime X! C'est le meilleur" ]}}}
```

Step 3: Predict

```
POST prediction/v1.1/query/mybucket%2Fmydata/predict

{ "data":{
    "input": { "text" : [
      "J'aime X! C'est le meilleur" ]}}}

{ data : {
  "kind" : "prediction#output",
  "outputLabel":"French",
  "outputMulti" :[
       {"label":"French", "score": x.xx}
     {"label":"English", "score": x.xx}
     {"label":"Spanish", "score": x.xx}]}}
```

Step 3: Predict

Apply the trained model to make predictions on new data

```
import httplib
# put new data in JSON format
params = { ... }
header = {"Content-Type" : "application/json"}

conn = httplib.HTTPConnection("www.googleapis.com")conn.
   request("POST",
   "/prediction/v1.1/query/mybucket%2Fmydata/predict",
   params, header)

print conn.getresponse()
```

Prediction API Capabilities

- Data
  - Input Features: numeric or unstructured text
  - Output: up to hundreds of discrete categories

- Training
  - Many machine learning techniques
  - Automatically selected
  - Performed asynchronously

- Access from many platforms:
  - Web app from Google App Engine
  - Apps Script (e.g. from Google Spreadsheet)
  - Desktop app

GOOGLE BIGQUERY

INTERACTIVE ANALYSIS OF LARGE DATASETS

Key Capabilities

- Scalable: Billions of rows

- Fast: Response in seconds

- Simple: Queries in SQL-like language

- Accessible via:
  - REST
  - JSON-RPC
  - Google App Scripts

Writing Queries

- Compact subset of SQL
  - **SELECT ... FROM ...**
    WHERE ...
    GROUP BY ... ORDER BY ...
    LIMIT ...;

- Common functions
  - Math, String, Time, ...

- Additional statistical approximations
  - TOP
  - COUNT DISTINCT

BigQuery via REST

- GET /bigquery/v1/tables/{table name}
- GET /bigquery/v1/query?q={query}

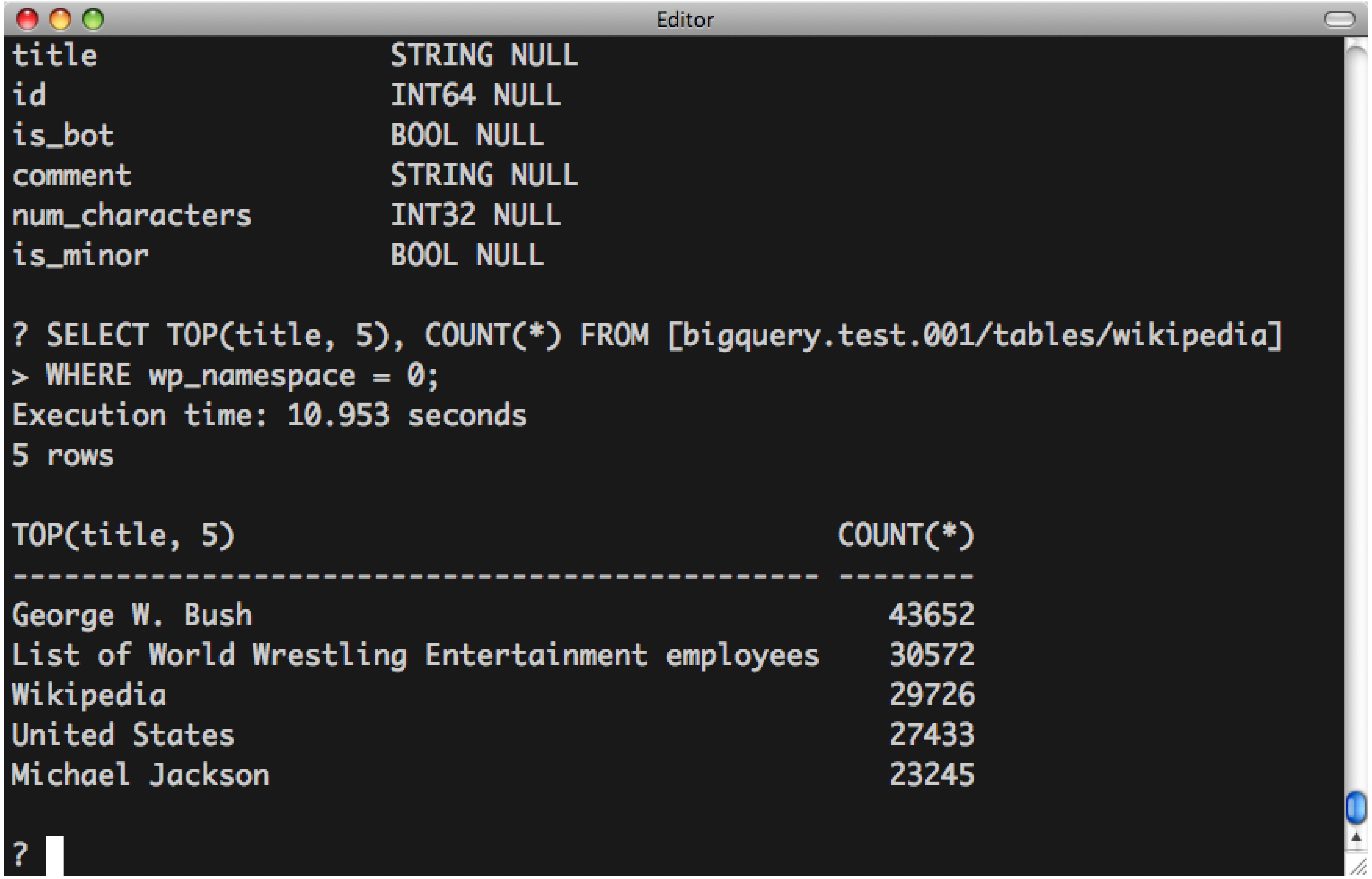
Sample JSON Reply:

```
{
  "results": {
    "fields": { [
      {"id":"COUNT(*)","type":"uint64"}, ... ]
    },
    "rows": [
      {"f":[{"v":"2949"}, ...]},
      {"f":[{"v":"5387"}, ...]}, ... ]
  }
}
```

Also supports JSON-RPC

Example Using BigQuery Shell

- Python DB API 2.0 + B. Clapper's sqlcmd
- http://www.clapper.org/software/python/sqlcmd/



```
title                 STRING NULL
id                    INT64 NULL
is_bot                BOOL NULL
comment               STRING NULL
num_characters        INT32 NULL
is_minor              BOOL NULL

? SELECT TOP(title, 5), COUNT(*) FROM [bigquery.test.001/tables/wikipedia]
> WHERE wp_namespace = 0;
Execution time: 10.953 seconds
5 rows

TOP(title, 5)                                   COUNT(*)
----------------------------------------------- --------
George W. Bush                                     43652
List of World Wrestling Entertainment employees    30572
Wikipedia                                          29726
United States                                      27433
Michael Jackson                                    23245

?
```

WORTH READING

SOME OF WHAT IS UNDER THE COVERS

FlumeJava – Processing Pipeline

A Java library that makes it easy to develop, test, and run efficient data-parallel pipelines. At the core of the FlumeJava library are classes that represent immutable parallel collections, each supporting a modest number of operations for processing them in parallel. Parallel collections and their operations present a simple, high-level, uniform abstraction over different data representations and execution strategies. To enable parallel operations to run efficiently, FlumeJava defers their evaluation, instead internally constructing an execution plan dataflow graph.

Sawzall – Parallel Data Analysis

We present a system for automating analyses. A filtering phase, in which a query is expressed using a new programming language, emits data to an aggregation phase. Both phases are distributed over hundreds or even thousands of computers. The results are then collated and saved to a file. The design -- including the separation into two phases, the form of the programming language, and the properties of the aggregators -- exploits the parallelism inherent in having data and computation distributed across many machines.

Dremel – Ad Hoc Query

A scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds. The system scales to thousands of CPUs and petabytes of data, and has thousands of users at Google.

Pregel – Comp Model for Graphs

Many practical computing problems concern large graphs. Standard examples include the Web graph and various social networks. In this paper we present a computational model suitable for this task. Programs are expressed as a sequence of iterations, in each of which a vertex can receive messages sent in the previous iteration, send messages to other vertices, and modify its own state and that of its outgoing edges or mutate graph topology. This vertex-centric approach is flexible enough to express a broad set of algorithms.

More information

– Google Storage for Developers
  - http://code.google.com/apis/storage

– Prediction API
  - http://code.google.com/apis/predict

– BigQuery
  - http://code.google.com/apis/bigquery

– Dremel
  - http://research.google.com/pubs/pub36632.html