# Caching and State Management

## Google App Engine

Google Cloud Platform

# Agenda

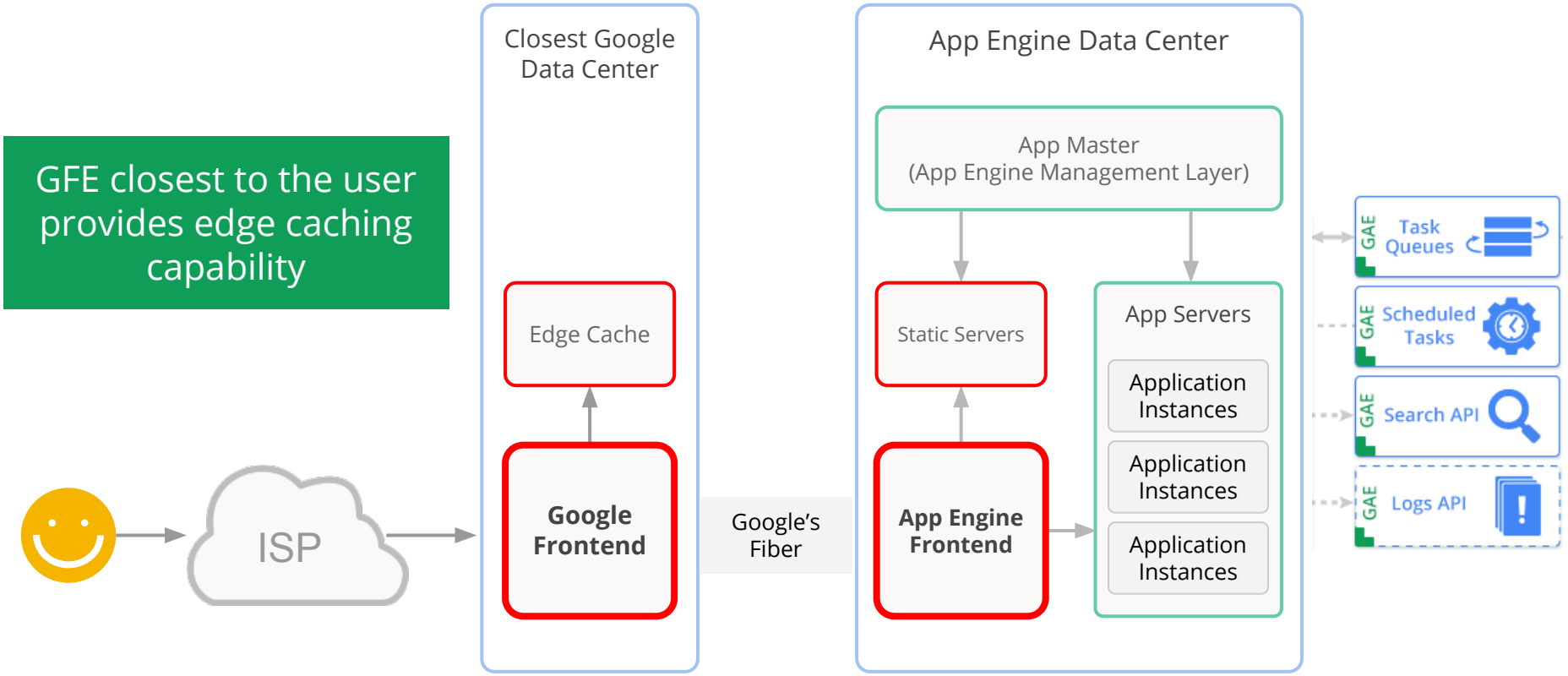
1. Edge Caching
2. Memcache Overview
3. Implementing Memcache
4. Caveats and Solutions

# Edge Caching

## Introduction

# Edge Caching at Google Frontends

GFE closest to the user provides edge caching capability



# Enable Edge Caching for App Engine Apps

## Option 1 - Use cache-control header on HTTP Response

```
class MyHandler(webapp.RequestHandler):
  def get(self):
    self.response.headers.add_header(
      'cache-control',
      'public, max-age=7200') # 2hr cache
```

May be honored by Google Frontend
(not guaranteed)

# Enable Edge Caching for App Engine Apps

## Option 2 - Define Static Content

### Java

appengine-web.xml

```
...
<static>
 <include path="/**.png" />
 <exclude path="/img/**.png />
</static>
...
```

### Python

app.yaml

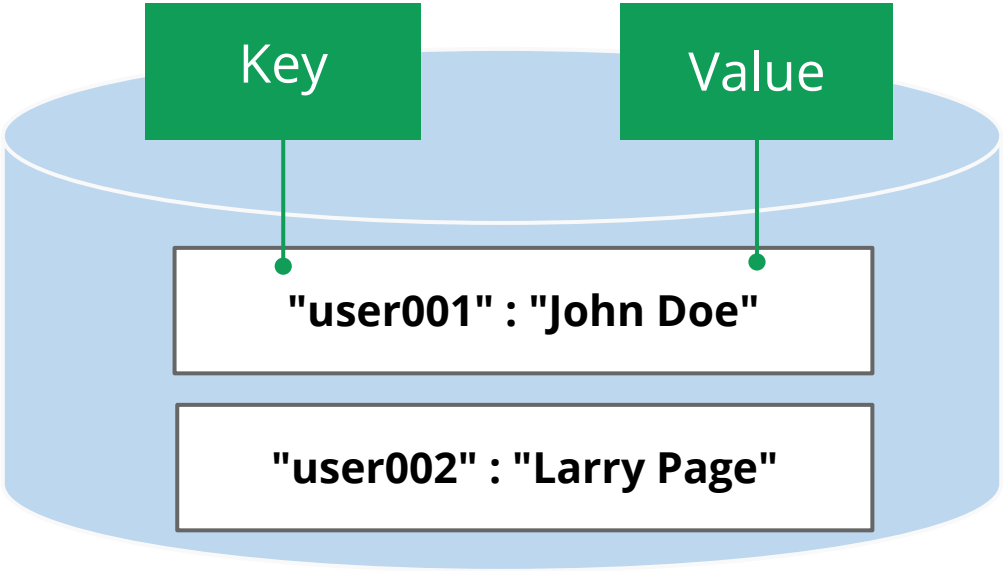
```
...
- url: /images
static_dir: static/images
OR
- url: /images/(.*)
static_files: static/images/\1
upload: static/images/(.*)
...
```

# Memcache

## Introduction

Google Cloud Platform

# What Is Memcache?



# Applications for Memcache

- Caching in front of Datastore
  - Cache entities for low-latency reads
  - Integrated into most ORM frameworks (ndb, Objectify)
- Caching for Read-heavy operations
  - User authentication token and session data
  - APIs call or other computation results
- Semi-durable shared-state cross app instances
  - Sessions
  - Counters/Metrics
  - Application Configurations

# Implementing Memcache

## Why Use Memcache?

Google Cloud Platform

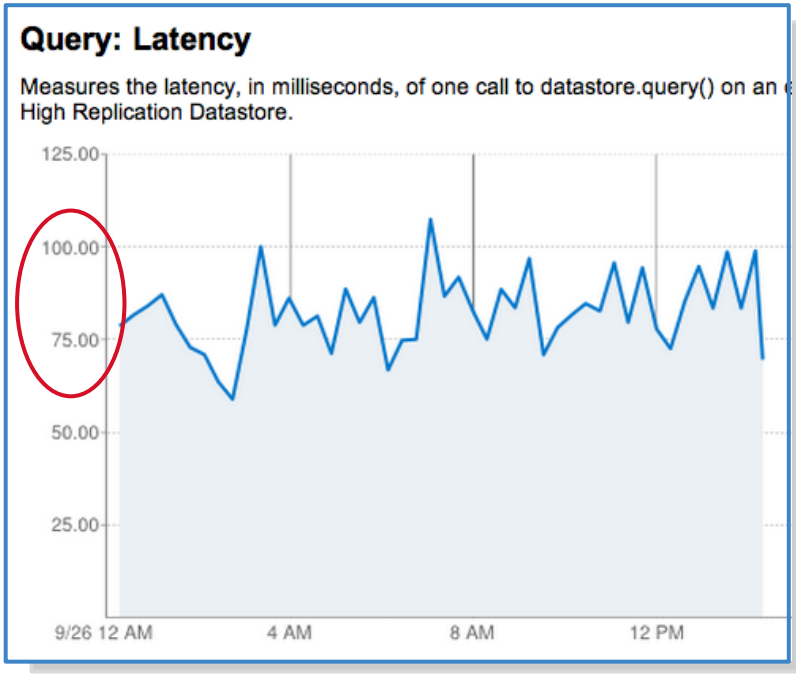
# Why Do We Use Memcache?

## Time and Money!

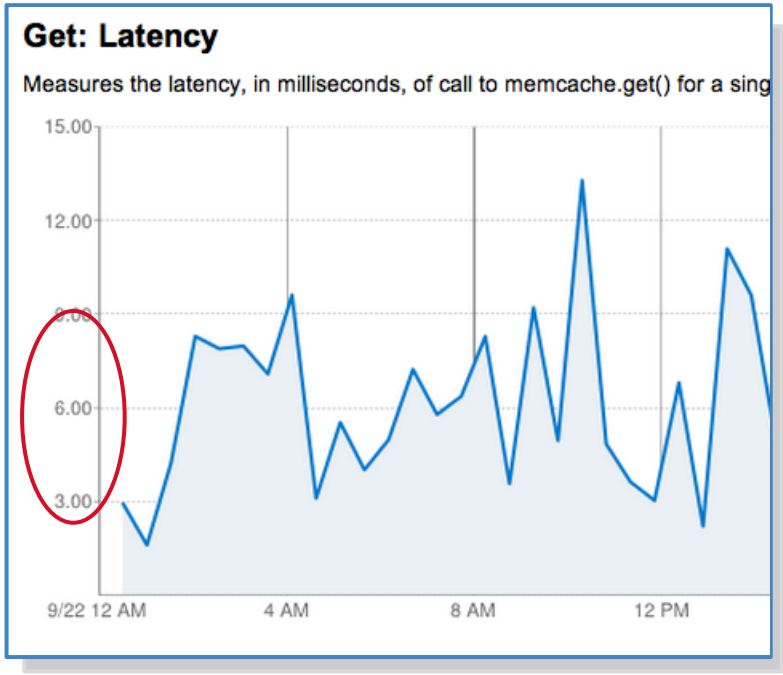
1. Improve Application Performance
2. Reduce Application Cost

# Memcache Speed

## How fast can Memcache be?




Datastore Query Latency




Memcache Read Latency

# Memcache APIs

## Java

- JCache APIs
- GAE Low-level Memcache APIs
- Objectify for Datastore

## Python

- `google.appengine.api.memcache` module
- `NDB` for Datastore

# General Memcache Usage Pattern

**Coordinate data *read* with Datastore**

- ❏ Check if Memcache value exists
  - If it does, display/use cached value directly; otherwise fetch the value from Datastore and write the value to Memcache

**Coordinate data *write* with Datastore**

- ❏ Update Memcache value
  - To handle race condition, leverage put if untouched/compare and set to detect race conditions
- ❏ Write the value to Datastore
  - Optionally, leverage the task queue for background writes

# Using GAE Client Library

python

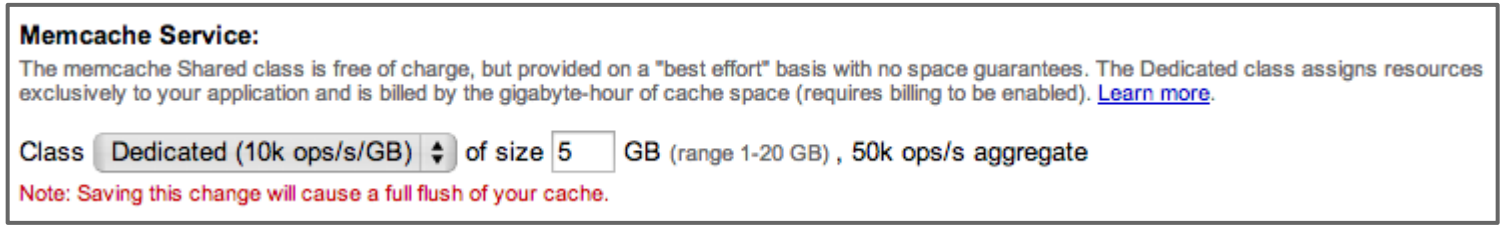
```
from google.appengine.api import memcache
...
value = memcache.get(key)

if value is None:
    value = get_value_from_db(key)
        if not memcache.add(key, value):
        logging.error('Memcache add failed.')
...
```

# Batch Operations

**`getAll (), putAll(), deleteAll()`**

- A single read or write operation for multiple memcache entries
- Further improve Memcache performance by reducing network traffic
- Batch size < 32 MB

# Dedicated Memcache




1. Provides fixed cache capacity that is exclusive to your application.
2. Billed by the GB per hour of cache size. Charged in 15 minute increments.
3. Gives you additional control over cache size. More predictable performance

**Note:** Whether shared or dedicated, Memcache is not a durable storage. Plan for your application to function without Memcache.

# Key Takeaways

1. Memcache is supported natively in GAE. Take advantage of it to improve your GAE application performance.

2. Memcache supports open standard JCache API. Many advanced features are available by GAE Memcache APIs to suit your application's need i.e. Batch, Atomic, Asynchronous operations

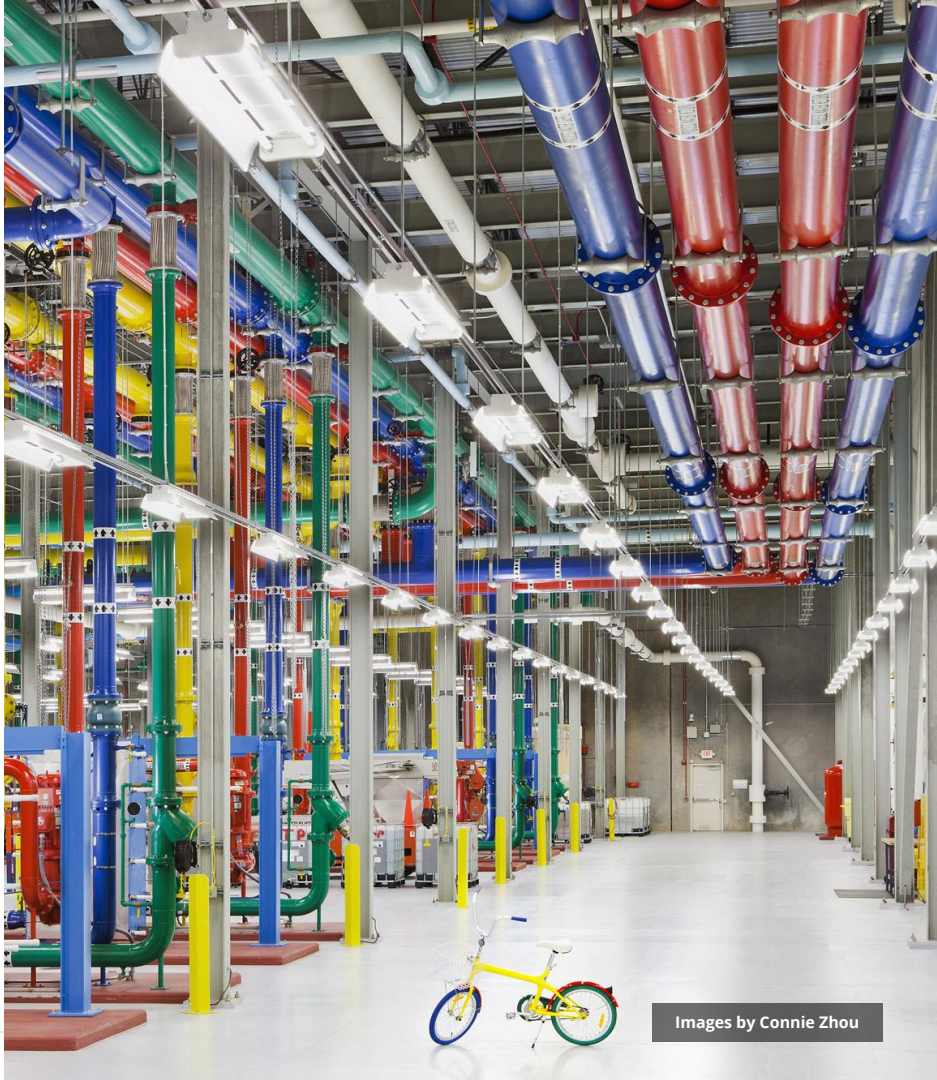
3. Seamless integration with GAE Datastore in a few libraries like Python ndb and Java Objectify

# Key Takeaways

4. Read-frequently and write-rarely data is most suitable in  combining with Memcache
5. Handle Memcache's volatility in your application
6. Use Memcache wisely, it is not an unlimited resource

# Hands-on

- You will create an App Engine application in either Java or Python
- You will then integrate the Memcache API to understand how it works



# Resources

- Memcache Java API Overview
  https://cloud.google.com/appengine/docs/java/memcache/

- Memcache Python API Overview
  https://cloud.google.com/appengine/docs/python/memcache/


cloud.google.com
Images by Connie Zhou