Introducing App Engine

Google App Engine

Google Cloud Platform

Agenda

1 Overview of App Engine

2 Design for Scalability And Reliability

3 App Engine Architecture

Google App Engine

Simple to Scale

- Autoscale

Easy to develop

- Free to start
- Build and test locally
- Focus on App Code

Trivial to manage

- Fully managed
- No patches/updates
- 24x7 operation by Google

Google App Engine

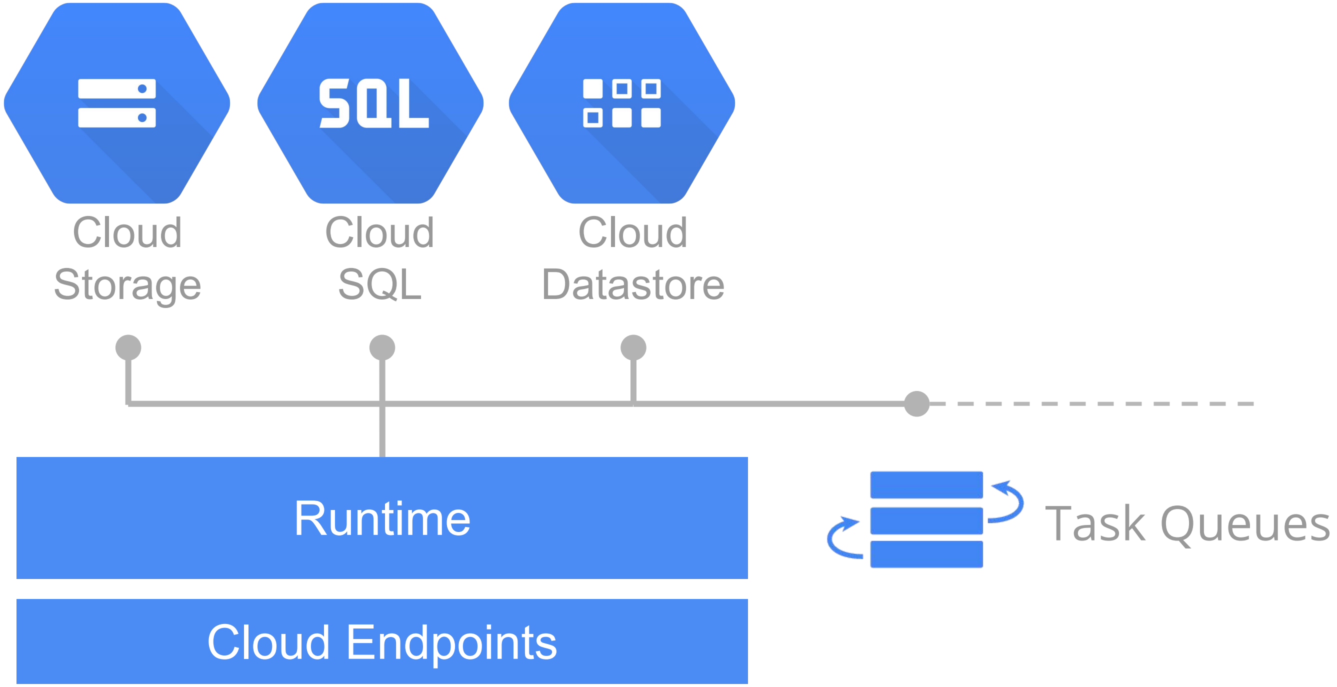
When to use?

- Focus on your code
- Managed environment
- Build micro-services

Capabilities

- Automatic resource provisioning
- Java, Python, PHP, & Go
- Identity, Memcache, Task Queue, Mail, and so on

Google App Engine

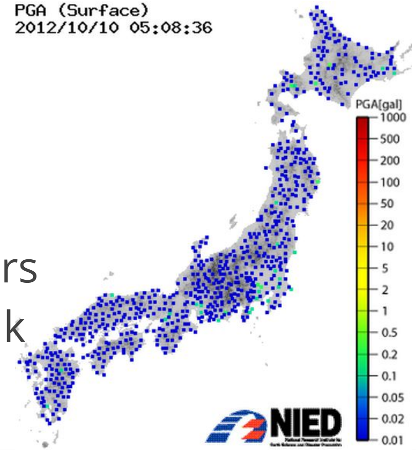


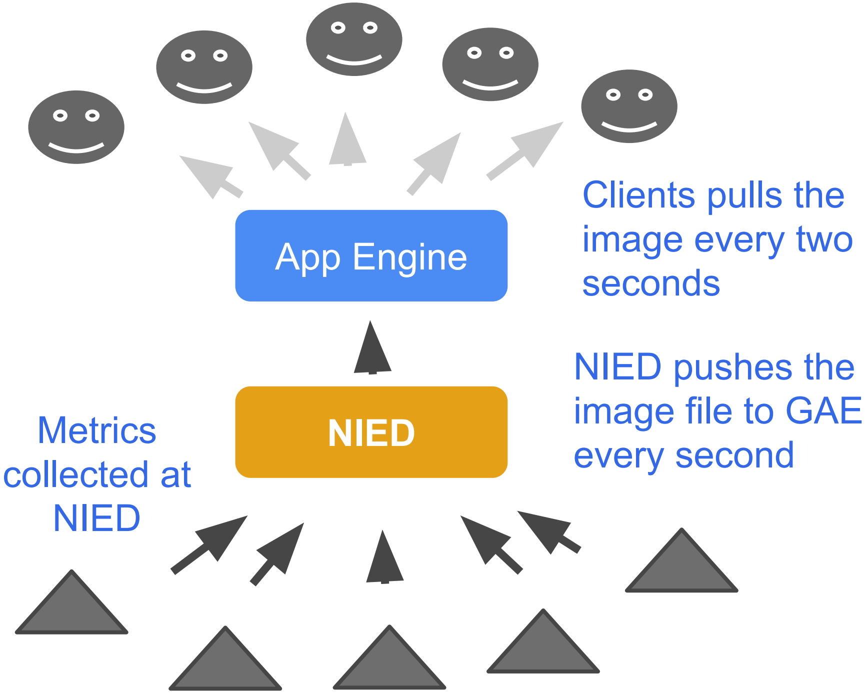
Design for Scalability And Reliability

Google App Engine

Use Case - Real Time Earthquake Monitor

- GAE Web App By NIED Japan + Google
  - National research Institute for Earth science and Disaster prev.
  - The blinking dots represents **real time Peak Ground Acceleration**
- YouTube video
  How it worked at the March 11, 2011 earthquake
- 20,000 concurrent users
  10,000 reqs/sec at peak





Clients pulls the image every two seconds

NIED pushes the image file to GAE every second

Metrics collected at NIED

Important Requirements

- **Scalability**
  - As it gets spike traffic at the event of earthquake
- **Reliability**
  - It is useless if it does not work at the time of disaster
- **Cost Effectiveness**
  - It would be too expensive to provision the system for peak traffic

Important Requirements

- **Scalability**
  - As it gets spike traffic at the event of earthquake
- **Re**
- **Co**

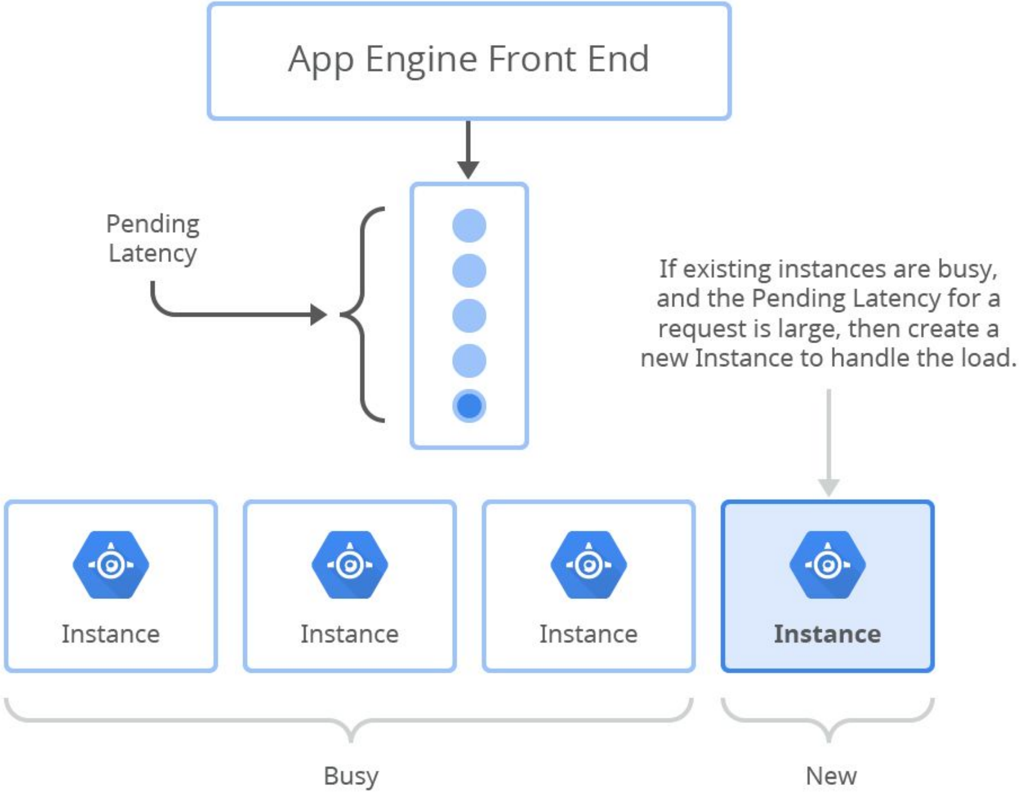
How do I design a web service that is **scalable**, **reliable**, and **cost effective** like this?

App Engine Application Scalability

App Engine watches **Pending Request Queue** of each *app version*

If your app receives a traffic spike:

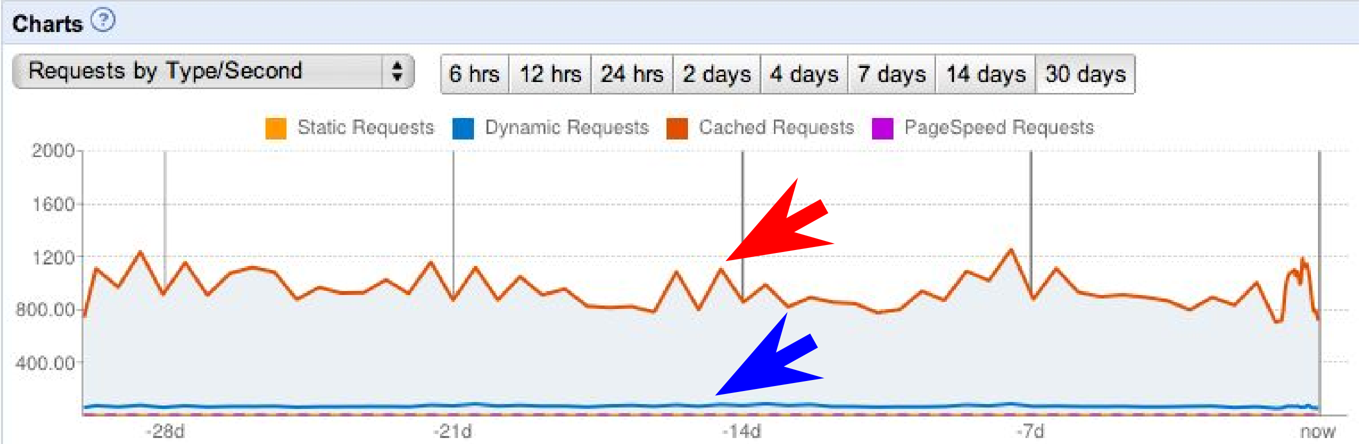
- Instances dynamically added/removed based on queue size



If existing instances are busy, and the Pending Latency for a request is large, then create a new Instance to handle the load.

Benefits of Autoscaling

Notice the huge difference between the traffic volume on the **red line** (the number of requests) and **blue line** (the number of requests handled by the application).



?

The Earthquake site broadcasts the image file to thousands of users **without handling all the requests by the app**. How does it do that?

Important Requirements

What About?

- ❑ Hardware Failures
- ❑ Traffic Spike
- ❑ Growing Data Storage
- ❑ Complex Design
- ❑ Complex Dev.
- ❑ Complex Admin
- ❑ Cost

App Engine Solves it:

- ✓ Hardware Failures
- ✓ Traffic Spike
- ✓ Growing Big Data
- ✓ Simpler Design
- ✓ Simpler Dev
- ✓ No Admin
- ✓ No Initial Funding

App Engine Design Goals

- **Empowers You to Design Your Applications in Google Way**
  - It is not just hosting
  - It is not just data center
- **Encapsulates Google's Best Practice for Scalability And Reliability**
  - Non-relational data model by Datastore/Bigtable
    - Sharding, Denormalization, and so on
  - Portable and service-oriented app design
    - Independent to each physical server or data center
    - Fast request handling that is horizontally scalable

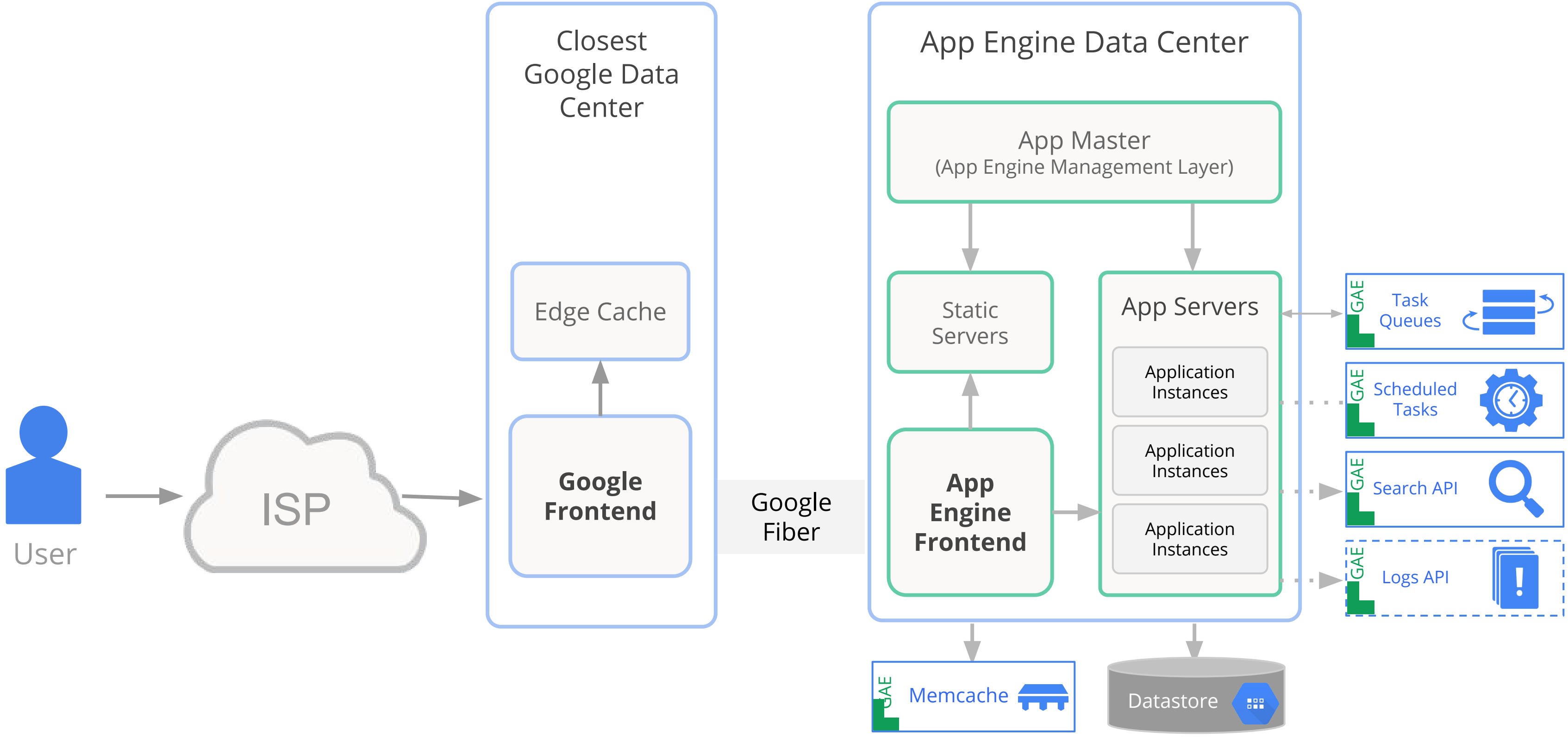
App Engine Design Goals ...

- **Significant Lower TCO - Total Cost of Ownership**
  - Economy of scale
  - Easy to develop and deploy
  - Free to start - no initial cost
  - Lower operation cost
    - No security patch, system upgrade etc
    - 24 x 7 operation by Google

App Engine Architecture

Google App Engine

App Engine Architecture



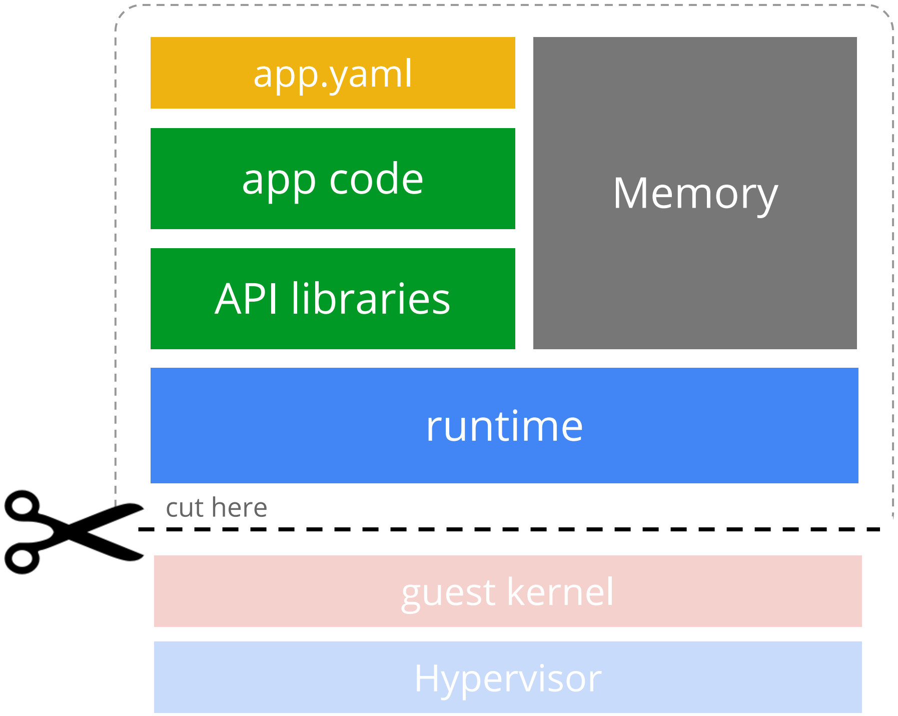
What Is An Application Instance?

Similar to a virtual machine, it provides a runtime environment for your application:

- **Dedicated memory** for your app
  - See: https://cloud.google.com/appengine/docs/python/modules/#instance_classes
- **Fully managed** sandbox
- No burden of managing OS; the overhead, device drivers, security, and so on

Unlike a virtual machine, App Engine can monitor user request processing and scale to match demand.

Anatomy of an instance



Key Takeaways

App Engine encapsulates all the costs and efforts

- Build and operate a web app with Google scale and reliability
- Encourage Google's best practices for scalability and reliability
- Significantly lower Total Cost of Ownership

Frontends provides scalability and reliability

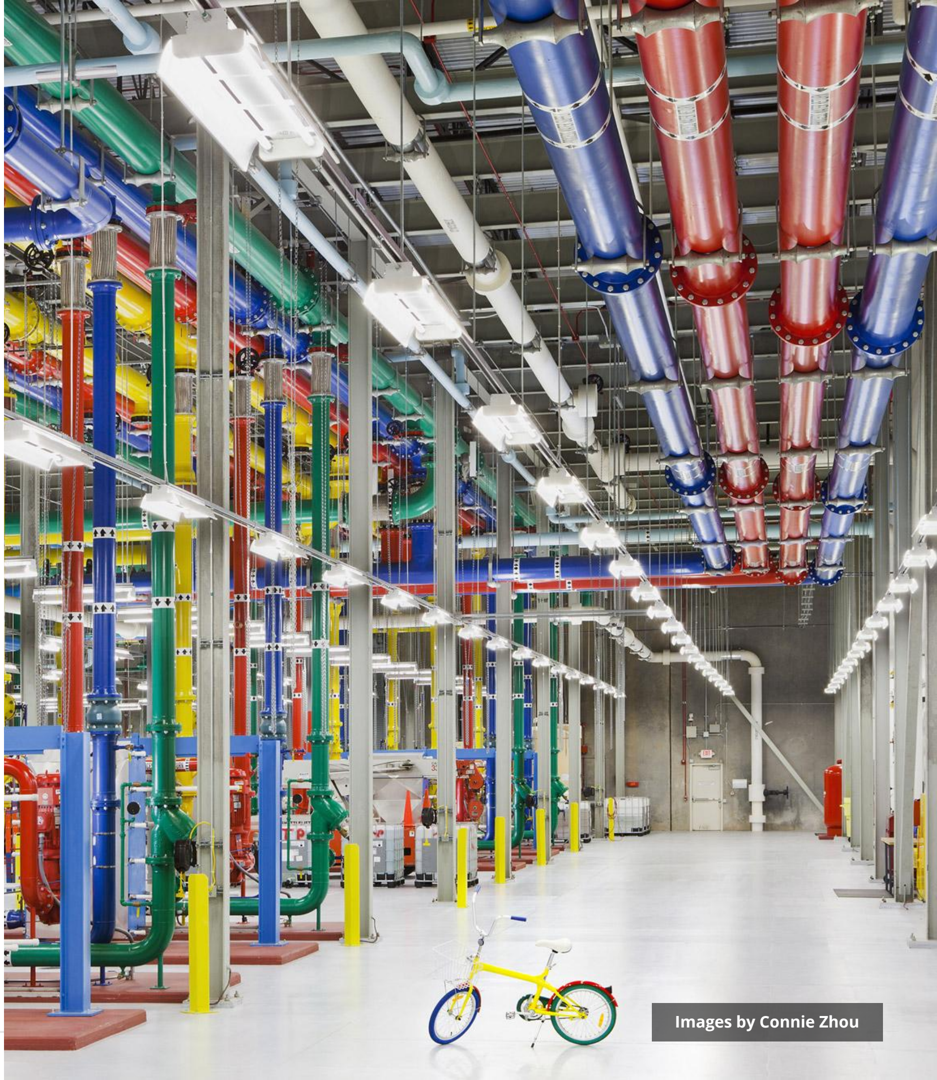
- *Google* Frontend
    - Edge Caching and Networking
- *App Engine* Frontend
    - Static Content delivery
    - Load Balancing

App Engine infrastructure manages scaling

- Automatic scaling of App Engine instances
- Manages Pending Latency and Idle Instances

Hands-on

- Deploy a sample application to Google App Engine



Images by Connie Zhou

Quiz

Which of the following **runtimes** is it possible to access from within App Engine using the features you've learnt about in this Module?
*(pick **one** answer)*

- ❑ Perl
- ❑ Node.js
- ❑ Ruby
- ❑ Java
- ❑ All of the above
- ❑ None of the above

Resources

- Google App Engine: Platform as a Service
  https://cloud.google.com/appengine/docs

- Overview of App Engine Features
  https://cloud.google.com/appengine/features/

- Google App Engine Articles
  https://cloud.google.com/appengine/articles/

cloud.google.com

Images by Connie Zhou