# Presentation about Platform as a service

Tobias Reincke

December 15, 2019

# Introduction / What is this Presentation as a Service?

service allowing development, management and running without building infrastructure

Typical for app development

shortened to PaaS

$\rightarrow$ Focus on Google App Engine

Is being delivered in 3 ways (definitions):

# Development & Uses

**– Public –**

consumer controlls deployment

minimal configurations

is provided:

A server and networks
B storage
C operating system
D middleware (like Java Runtime, .NET)
E database services
F other services
G uses virtual machines host and scale services

# Development & Uses

– **Private** –

behind firewall

A as software

B as appliance

– **Extension of IaaS** –

with software added on top

# A short History

First PaaS : Zimski's Launch in 2006

- javascript end to end development
- virtual servers
- configuration,
- security
- and backups
- pay you go

Google App Engine Launch 2008

Launch of Microsoft Azure as Windows Azure in February 2010

here we are (after 10 years of steady updates)

# Google App Engine

Google
App Engine

# Google App Engine

- serverless platform
- developing & hosting web applications
- several languages , libraries, frameworks
- scaling on demand

# First: Supported Languages & APIs

Python 3 (.7) & Python 2

JVM languages (Java (8), Kotlin, etc.)

support for DotNet-Core's CVM (C#)

commonly used scripting languages:

- Ruby
- Node.js (javascript)
- Go
- PHP

Rest APIs

No C or C++

No C-modules to integrate with python

# Pay as You Go

pay as you host a server

| Instance class | Cost per hour per instance |
|---|---|
| B1 | $0.05 |
| B2 | $0.10 |
| B4 | $0.20 |
| B4_1G | $0.30 |
| B8 | $0.40 |
| F1 | $0.05 |
| F2 | $0.10 |
| F4 | $0.20 |
| F4_1G | $0.30 |

Figure:

Every feature is calculated on its own.
$\longrightarrow$ will be mentioned in their respective parts

# Database Features

> supports for SQL and MySQL and NOSQL

Cloud BigTable

BigQuery

Cloud SQL

GQL

# Big Query

## Definition

> Big Data analysis (gigabyte to petabyte) using Ansi SQL

- making it fast as heck
- no primary key support $\rightarrow$ have to sort out duplicate data as a workaround (still fast)
- Google does not mention this issue
- takes upload streams, queries then stores In Google Storage
- prices for those upload are quite expensive(200 Mb per 1 ct)
- 3 basic operations:
  - Loading data into a table
  - Copying data into a table (the same but without load)
  - write query results to a table
- flatrate and subscriptions
  $\rightarrow$ recommended subscriptions since query results are unknown
- can use UML & DDL

| Operation | Pricing | Details |
| --- | --- | --- |
| Active storage | $0.020 per GB | The first 10 GB is free each month. See Storage pricing for details. |
| Long-term storage | $0.010 per GB | The first 10 GB is free each month. See Storage pricing for details. |
| BigQuery Storage API | $1.10 per TB | The BigQuery Storage API is not included in the free tier. |
| Streaming Inserts | $0.010 per 200 MB | You are charged for rows that are successfully inserted. Individual rows are calculated using a 1 KB minimum size. See Streaming pricing for details. |
| Queries (on-demand) | $5.00 per TB | First 1 TB per month is free, see On-demand pricing for details. |
| Queries (monthly flat-rate) | $10,000 per 500 slots | You can purchase additional slots in 500 slot increments. For details, see Monthly flat-rate pricing. |
| Queries (annual flat-rate) | $8,500 per 500 slots | You can purchase additional slots in 500 slot increments. You are billed monthly. For details, see Annual flat-rate pricing. |

# Data Store

- Part of Google Clouds DataStore -exists to save object oriented Datasets and Entities in NoSQL
  - writes fast
  - provides SQL-like queries
  - indexes
  - ACID (atomicity, consistency, isolation, durability) $\rightarrow$ Data safety guaranteed
  - mobile development
  - user profiles
  - product catalogues and inventory list and simple things like that
  - Groups of Objects, Objects, Properties, Unique ID as Key

# Data Store

Los Angeles

| | Free limit per day | Price above free limit (per unit) | Price Unit |
|---|---|---|---|
| Stored data | 1 GB storage | $0.108 | GB/Month |
| Entity Reads | 50,000 | $0.036 | per 100,000 entities |
| Entity Writes | 20,000 | $0.108 | per 100,000 entities |
| Entity Deletes | 20,000 | $0.012 | per 100,000 entities |
| Small Operations | 50,000 | Free | - |

# Cloud storage

for files and pictures, works like any file manager, can be accessed via conventional code

- standard for frequent access rate
  - 2 ct up to 3.6 ct per GB
- nearline

  1 up to 2 ct per GB
- coldline

  0.4 ct up to 0.9 ct per GB

# App Engine Console

$ gcloud app <command>

- browse
- create
- deploy
- describe
- open-console

# Memcache

But what does it do?

# Mem(e)cache

## But what does it do?

literally a fucking memory cache

1

[1] This page is just the conclusion of me messing with latex a bit and memeing. Of cause it won't be included in the final Form. Or will it be ?

# Memcache

**Shared Memcache:**

it is shared between all users

**Dedicated Memcache:**

Google offers up to 20 GB of memory cache for often used returns (100 on us-central!)

as long as free it saves every request

250 Byte key length (hashed if bigger)
$\rightarrow$ more than enough

separate set and get-hit

6 ct per GB/hour

writes at 5k items and reads 10k per GB/s

# Restrictions

Apps can only use Google intern filesystems (being the ones presented)

only the aforementioned languages can be run, no c (or pyrex) modules for python code

App Engine is only able to execute code called from an HTTP request

# Restrictions

Apps can only use Google intern filesystems (being the ones presented)

only the aforementioned languages can be run, no c (or pyrex) modules for python code

App Engine is only able to execute code called from an HTTP request

BUT:

- executed code can be called by itself via scheduled https requests as a workaround

No sticky sessions.

No Maven for Java.

# Pro

Flexible pricing

LOTS of potentially useful feature you don't have to integrate yourself

No need to buy servers or server space (no maintenance)

Makes solving the problem of scaling easier

Free up to a certain level of consumed resources

Suitable for apps that just store and retrieve data.

$\rightarrow$ Nice option for start-ups/individuals $\leftarrow$ only need for a google account

better peak performance than Amazon's EC2 service (more overall CPU hours)

# Contra

No (easy) transferiability/migration.

App Engine requires developers to use only its supported languages, APIs, and frameworks

There are open source project trying to fix this.

Don't count on open source.

bulk downloads only for python (excluding java, php, go etc.)

Not suitable for CPU intensive calculations. They are slower and expensive.

# Contra

Suffers from traditional PaaS problems

relatively long development time for something rather simple, it has to be simple

if an app makes profit at Google scale then it probably makes enough money to run on its own servers $\longrightarrow$ scalability not worth

lots of minor limitations making deep data analysis difficult

sudden traffic spikes need a lot of inactive servers online $\longrightarrow$ \$\$\$ for dealing with those

# Upper Limits of Using it for free [2]

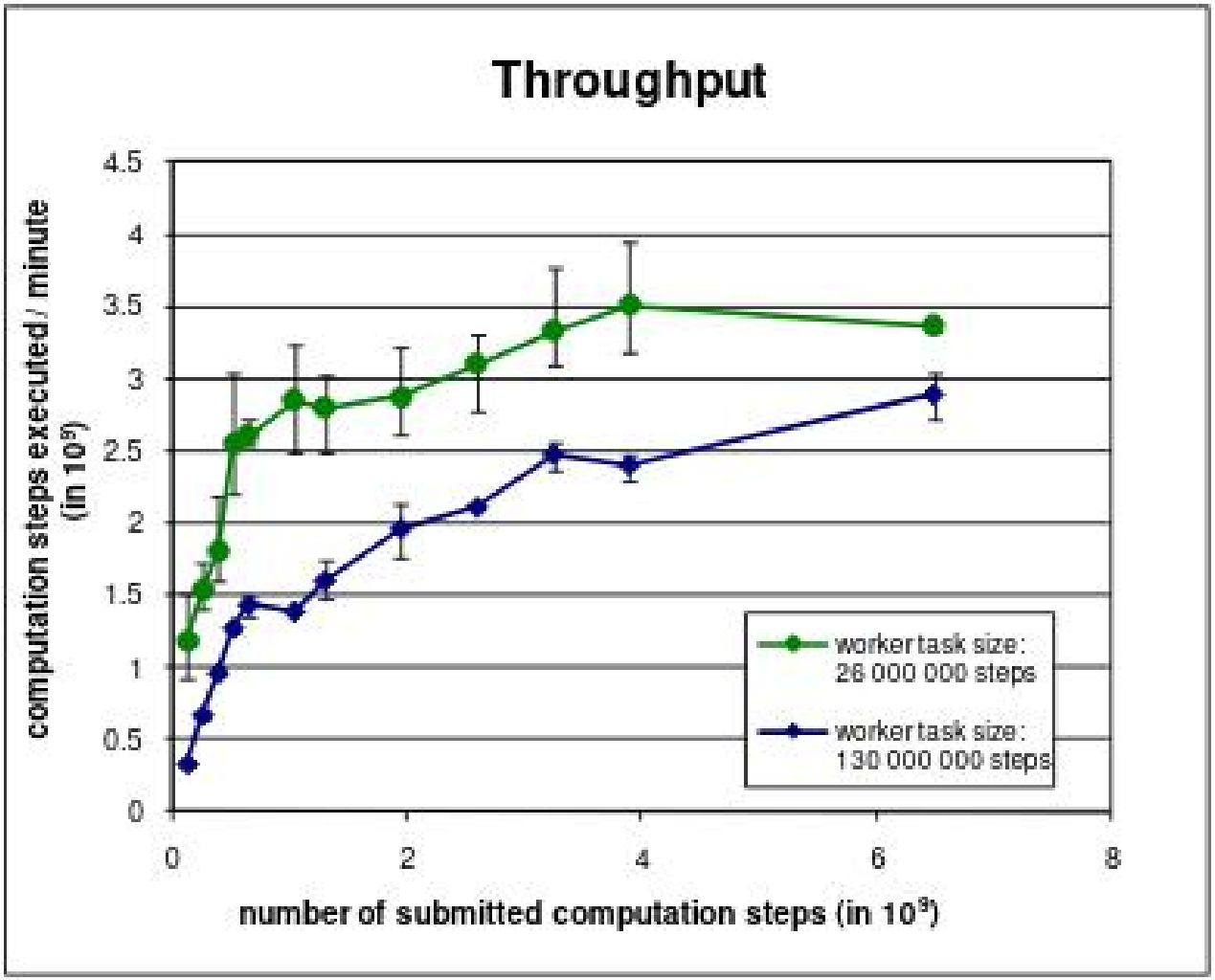
Researchers used Taskline API to measure Google App Engine in great detail

Pipeline and Map API to build Framework to schedule and split task

Used average computational steps / min to benchmark since amount of computational nodes is unknown/-predictable

[2] https://www.icsr.agh.edu.pl/ malawski/google-appengine-ieee-2011.pdf

# Upper Limits of Using it for free [3]




[3] https://www.icsr.agh.edu.pl/ malawski/google-appengine-ieee-2011.pdf

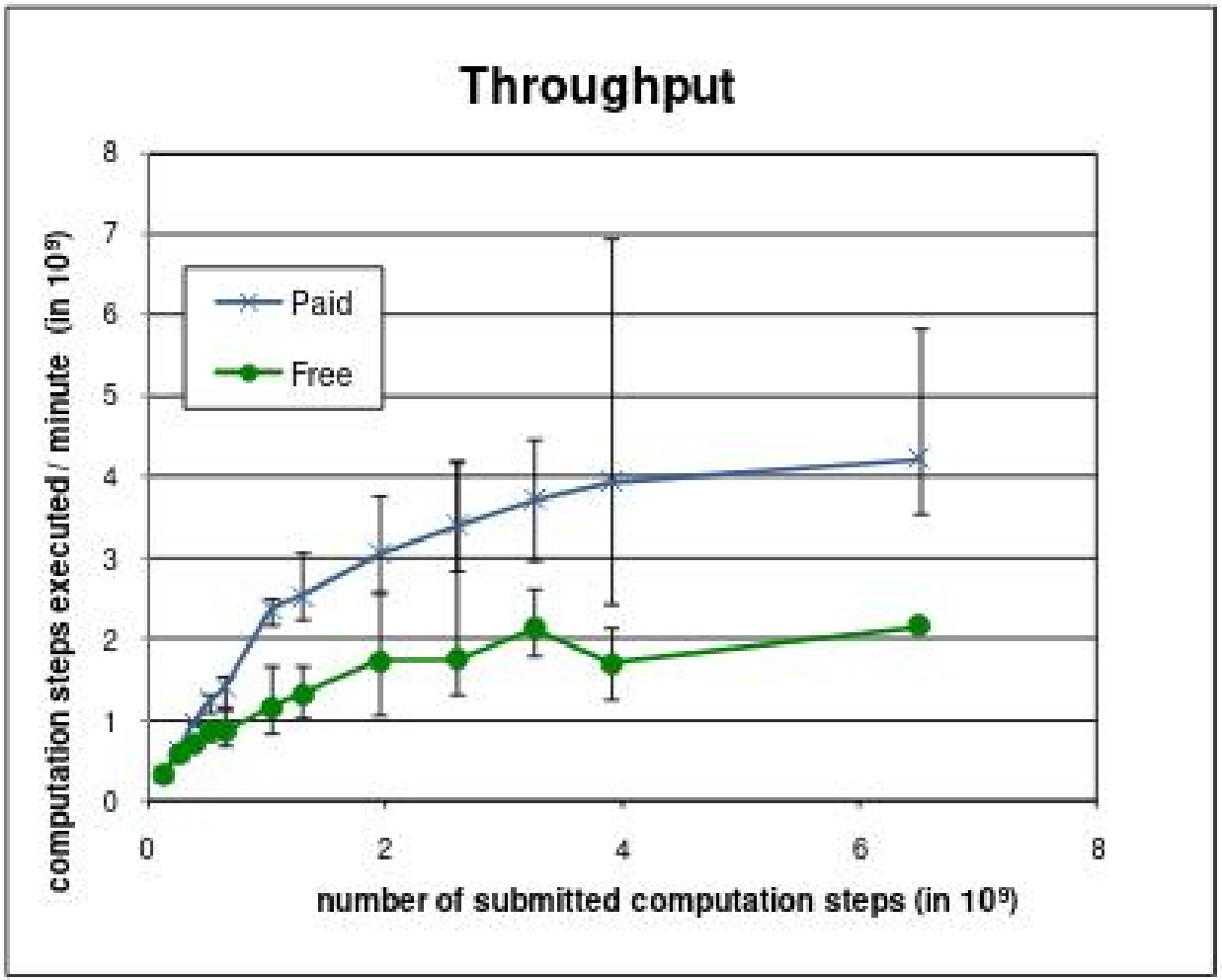

Throughput
computation steps executed / minute (in $10^9$)
number of submitted computation steps (in $10^9$)
Paid
Free
0
1
2
3
4
5
6
7
8
0
2
4
6
8

Table 1
Comparison of free computing capabilities of major cloud providers. Peak speedup computed vs 2.16 GHz PC (laptop) processor.

| | App Engine | Amazon EC2 |
|---|---|---|
| Free period | Unlimited | 1 year |
| CPU share | 27% daily | 5% monthly |
| Peak speedup | 10 | 2 |

# Kudos

To..

- Wikipedia
- stack overflow