Basics of Cloud Computing – Lecture 6

Platform as a Service (PaaS)

Satish Narayana Srirama

Several slides are taken from Pelle Jakovits

Google Cloud Platform
Outline

• Introduction to PaaS
• Google Cloud
• Google App Engine
• Other PaaS providers
• Advantages & disadvantages of PaaS
Background

• Until now we have discussed mostly IaaS
• IaaS mainly provides virtual machines and resources
  – User need not have to purchase the hardware
  – IaaS can make better use of resources (utilization)
• You already worked with OpenStack instances on SciCloud
Troubles with IaaS

• Need a running environment or a development and testing platform
  – To design applications or services

• Users still require automatic decision-making of dispatching of jobs to available resources
  – We already have discussed solutions for loadbalancing and auto-scaling

• Complete system administration and monitoring falls on the user’s shoulders
Cloud Services

SaaS

PaaS

IaaS

http://nolegendhere.blogspot.com.ee/2012/06/presentation-4-5-7.html
Platform as a Service - PaaS

• Complete platform for hosting applications in Cloud
• All the infrastructure is managed for you
• Enables businesses to build and run web-based, custom applications in an on-demand fashion
• Eliminates the complexity of selecting, purchasing, configuring, and managing hardware and software
• Dramatically decreases upfront costs
PaaS Characteristics

- Multi-tenant architecture
- Built-in scalability of deployed software
- Integrated with web services and databases
- Users are provided with tools to simplify creating and deploying applications
- Simplifies prototyping and deploying startup solutions
PaaS Characteristics - continued

• Users only pay for services they use
• More fine grained cost model
• Provides tools to handle billing and subscription management
• Typically introduces vendor lock-in
Types of PaaS

• **Standalone Application Platforms**
  – Typically built on top of an existing IaaS
  – Provides development tools for designing and deploying software
  – Provide all required computing resources and services needed for hosted applications

• **Social Application Development Platforms**
  – Used to develop add-ons and internal applications for social websites such as Google+ and Facebook
  – Integrated API with the social website platform
  – Can be seen as extending SaaS platform
Types of PaaS - continued

• Open-Computing Platforms
  – Not tied to a single IaaS provider
  – Goal is interoperability and supporting open-source platform tools
  – Supports applications that are written in numerous languages and that use any type of database, operating system, and server
  – E.g. Cloud Foundry, Red Hat OpenShift etc.
Google Cloud

• IaaS & PaaS services
• All services built on top of Google hardware located across the globe
  • Global redundancy and service locality
• Utilizes Google’s Global network
  – Dedicated fiber optic networks
  – Even between continents
  – Multilevel location based data caching
• Google Managed Services
  – Database administration, server configuration, sharding and load balancing managed for the user

https://cloud.google.com/
Google Cloud Services

• Compute Engine
  – IaaS providing virtual machines similar to Amazon EC2

• App Engine
  – PaaS for directly hosting applications

• Container Engine
  – Powerful cluster manager and orchestration system for running your Docker containers
    • Docker uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces
    • Independent "containers" run within a single Linux instance
    • Avoids the overhead of starting and maintaining virtual machines
Google Cloud Services - Storage

- Cloud Storage
  - Object storage for applications
- Cloud SQL
  - Fully managed Relational MySQL
  - beta support for PostgreSQL
- Cloud Datastore
  - Automatically scaled NoSQL storage
- Cloud BigTable
  - NoSQL database for warehousing
  - A high performance NoSQL database service for large analytical and operational workloads
Google Cloud Services - continued

• Big Data
  – Big Query
    • SQL like queries against multi-terabyte/petabyte datasets
  – Cloud Dataflow
    • Managing data processing workflows
    • Supports both stream and batch execution of pipelines
  – Cloud DataProc
    • Managed data processing with Hadoop and Spark

• Cloud Endpoints
  – Create RESTful services to make your code accessible from iOS, Android and Javascript clients
Free Cloud Account

• Apply at https://cloud.google.com/free-trial/
• $300 Credit for 12 months
• Maximum limit of 8 cores at a time for instances
• Need credit card to sign up
• Won't be billed until upgrade to paid account
• Only the portion of usage above the App Engine free daily quota is charged against credit
• Several other services are also available for free
Google App Engine
Google App Engine

- PaaS for developing and hosting web applications in Google-managed data centers
- Easy to build, maintain, and scale applications
- No servers to maintain or configure by yourself
- Upload & Go
- Was created before Google Cloud became available for public use
- Supported languages
  - Python, Java, PHP, Go
Deployment Life Cycle

- Write Code
- Test Locally
- Admin via Web Console
- Push to Google servers
App Engine Characteristics

- **Persistent storage** with queries, sorting, and transactions
- App Engine distributes user requests across multiple servers and **scales servers** to meet **dynamic traffic** demands
- **Asynchronous task queues** for performing work outside the scope of a request
- **Scheduled tasks** for triggering events at specified times or regular intervals
- **Integration** with all other Google Cloud services and APIs
- Your application runs within its own **secure, sandboxed and reliable** environment
  - Independent of the hardware, operating system, or physical location of the server
Use Cases

• Rovio
  – Transition “Angry Birds” games to an online environment
  – Use a platform that could support explosive demand
  – Easily add new features to improve the user experience

• Ubisoft
  – Develop a web-based version of a console game “From Dust”
  – Find an easy-to-use system so developers focus on the user experience
  – Scale effortlessly to accommodate a fast-growing number of players

• Best Buy
  – Quickly develop and deploy apps to achieve its business goals
  – Create apps that are scalable and low-maintenance
  – Dramatic time and cost savings in app development, which has led to the creation of many more apps
Services of Relevance for App Engine

- **Google Cloud SQL** - A fully-managed web service that allows you to create, configure, and use relational databases in Google's cloud

- **Datastore** - A schemaless object datastore providing robust, scalable storage for your web application, a rich data modeling API, and a SQL-like query language called GQL

- **Blobstore** - Allows your application to serve large data objects, such as video or image files, that are too large for storage in the Datastore service
Other data services

• **Search** - Allows your application to perform Google-like searches over structured data such as: plain text, HTML, atom, numbers, dates, and geographic locations.

• **Memcache** - A distributed, in-memory data cache to improve application performance

• **Logs** - Provides programmatic access to application and request logs from within your application
Communication

- **Channel** - Creates a persistent connection between your application in Google servers and JavaScript clients so you can send messages to clients in real time without "polling"
- **Mail** - Sends email messages on behalf of administrators and users with Google Accounts, and receives mail at various addresses
- **XMPP** - Enables an application to send and receive chat messages to and from any XMPP-compatible chat messaging service
- **Traffic Splitting** - Allows you to roll out features for your app slowly over a period of time. Traffic Splitting works by splitting incoming requests to different versions of your app
Process management

- **Task Queue** - Allows applications to perform work outside of a user request, and organize that work into small, discrete units, called "tasks," to be executed later.

- **Scheduled Tasks** - Allows applications to configure regularly scheduled tasks that operate at defined times or regular intervals.

- **Remote** - Lets external applications transparently access App Engine services. For example, you can use Remote API to access a production datastore from an app running on your local machine.
Other services

- **App Identity** - Gives code access to the application identity; provides framework to assert this identity over OAuth.
- **Users** - Allows applications to sign in users with Google Accounts or OpenID, and address these users with unique identifiers.
- **Capabilities** - Provides detection of outages and scheduled maintenance for specific APIs and services, so that your application may bypass them or inform your users.
- **Multitenancy** - Makes it easy to compartmentalize your data to serve many client organizations from a single instance of your application.
- **PageSpeed** - A family of tools that automatically optimizes the performance of your application.
IaaS Pricing Model

- Per instance hour
- Per storage amount/month
- Per Network bandwidth/month
- Per additional services
  - Static IP
  - Autoscaling/load balancing
## PaaS Pricing Model

<table>
<thead>
<tr>
<th></th>
<th>FREE LIMIT PER DAY</th>
<th>PRICE ABOVE FREE LIMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instances</strong></td>
<td>28 instance hours</td>
<td>$0.05 / instance / hour</td>
</tr>
</tbody>
</table>
| **Cloud Datastore (NoSQL)** | • 50k read/write/small  
                         | • 1 GB storage                                     | • $0.06 / 100k read or write ops 
                         |                                                 | • Small operations free* 
                         |                                                 | • $0.18 / GB / month |
| **Network Traffic (Outgoing)** | 1 GB               | $0.12 / GB             |
| **Network Traffic (Incoming)** | 1 GB              | FREE                   |
| **Cloud Storage**    | 5 GB               | $0.026 / GB / month    |
| **Memcache**         | • Free Usage of Shared Pool  
                         | • No free quota for Dedicated Pool                | • Free Usage of Shared Pool 
                         |                                                 | • Dedicated Pool: $0.06 / GB / hour |

Data taken in 2016  
[https://cloud.google.com/products/calculator/](https://cloud.google.com/products/calculator/)
# PaaS Pricing Model

<table>
<thead>
<tr>
<th>Service</th>
<th>FREE</th>
<th>LIMIT PER DAY</th>
<th>PRICE</th>
<th>ABOVE FREE LIMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td></td>
<td>- 1000 basic operations</td>
<td></td>
<td>- 0.50 / 10k searches</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0.01 GB indexing documents</td>
<td></td>
<td>- 2.00 / GB indexing documents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0.25 GB document storage</td>
<td></td>
<td>- 0.18 / GB / month Storage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 100 searches</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email API</td>
<td>100</td>
<td>recipients</td>
<td></td>
<td>Contact Sales</td>
</tr>
<tr>
<td>Logs API</td>
<td>100 MB</td>
<td></td>
<td></td>
<td>$0.12 per GB</td>
</tr>
<tr>
<td>Task Queue</td>
<td>5 GB</td>
<td></td>
<td></td>
<td>$0.026 / GB / month</td>
</tr>
<tr>
<td>Logs Storage</td>
<td>1 GB</td>
<td></td>
<td></td>
<td>$0.026 / GB / month</td>
</tr>
<tr>
<td>SSL Virtual IPs</td>
<td>-</td>
<td></td>
<td></td>
<td>$39 / virtual IP / month</td>
</tr>
<tr>
<td>Bundled Services</td>
<td></td>
<td>Cron, Image Manipulation, SNI SSL Certificates, Socket API, Task Queue API, URLFetch, Users API</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Windows Azure

• Hybrid PaaS & IaaS cloud platform
• Designed more for enterprise applications
• Programming languages
  – .NET, Java, PHP, Node.js, Python, or Ruby
• Datastores
  – Azure SQL database and NoSQL storage

https://azure.microsoft.com/
Windows Azure

• BigCompute
  – HPC on demand
  – MPI applications with Remote Direct Memory Access (RDMA)

• HDInsight
  – Setting up dynamic Hadoop clusters for Data Analysis

• Examples of applications running in Azure:
  – Office 365, Skype, Bing, and Xbox
  – WebZen, Toyota Gazoo.com, HALO
AWS Elastic Beanstalk

- Languages: Java, .NET, PHP, Node.js, Python, Ruby, Go
- Platforms: Docker, Apache, Nginx, Passenger, and IIS
- Automatically handles deployment, capacity provisioning, load balancing, auto-scaling, application health monitoring
- More manual control available (and required)

https://aws.amazon.com/elasticbeanstalk/
Other PaaS Examples

• AppScale  [https://www.appscale.com/]
  – Open-Source framework
  – Supports Google App Engine Applications
  – Supports MySQL Clusters, HBase, Hypertable, and Apache Cassandra
  – Python, Go, and Java applications

• Cloud Foundry  [https://www.cloudfoundry.org/]
  – open-source PaaS platform that supports programming languages Java and Scala

• AppFog  [https://www.ctl.io/appfog/]
  – Supports Java, Node, .Net, Ruby, PHP, MySQL, Mongo, PostgreSQL
  – Choose your own Cloud Provider.

• Heroku: Cloud Application Platform  [https://www.heroku.com/]
  – Was one of the first PaaS services offered on the market
  – Supports Ruby, Java, Scala, and Python applications
PaaS Advantages

- User does not have to manage low level computing resources and services
- Many services ready to use in a plug-in fashion without any configuration or setup
- **Provider handles most of the non functional requirements of your applications**
- **Scaling is automatically managed** by the platform
- Easier and Agile application deployment
  - Simplifies prototyping and application startups
- **Lower costs**
  - Pay for only what is used
  - More fine-grained cost model than in IaaS
- Platform provider has the best knowledge to optimize the services running on the underlying hardware
PaaS Disadvantages

• **Not in full control** over:
  – Computing resources (Intel vs AMD, GPU’s, FPGA, …)
  – Software and library versions
  – Service configuration
• Available programming languages are often limited
• **Vendor lock-in**
• Offered services may not be flexible enough for user needs
• Have to **fully trust** in the PaaS provider
  – Billing accuracy
  – Security
  – Reliability
  – Data ownership
• What happens when application exceeds billing quotas? What happens when payments fail?
That's All

• This weeks practice session is:
  – Creating and deploying Google App Engine applications using python

• Next lecture
  – Other Cloud Services & Cloud based research at Mobile & Cloud Lab
References

• Mastering Cloud Computing: Foundations and Applications Programming
  Authors: Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi

• Google Cloud Platform https://cloud.google.com/

• Chakkrit Tantithamthavorn, Introduction to Google App Engine,
  https://www.slideshare.net/klainfo/introduction-to-google-app-engine-13223789
AppEngine for Mobiles

• Define business logic on App Engine and access them via RESTful APIs on multiple platforms including Android, IOS and JavaScript.
• Deploy an app in 300 seconds
• Automatically generate strongly-typed client libraries for iOS, Android, and JavaScript.
• Geo-proximity search
• Push Notifications to iOS and Android
• Simple access to data storage and processing
AppEngine for Mobiles