

Functional Analysis of Platform as a Service (PaaS)

Gagana P¹, Vaishak Sundares²

UG Student, Department of CSE, Acharya Institute of Technology¹

Asst. Professor, Department of CSE, Acharya Institute of Technology, India²

Abstract: Cloud computing is a promising approach for the efficient use of computational resources. It is quickly revolutionizing software development, deployment, and lifecycle management. Cloud Computing is applied to geographically distributed data centres. Clouds serve Developers or Businesses, or large Enterprises as a way to host applications. Such applications are delivered to client PC's or PC-like smart phones. We consider the case where the end-users are in need of a highly efficient, responsive, and very functional interface that can dynamically load the pre-existing web portals from the cache rather than to load from the internet itself, operable by mobile devices. We explored the resultant needed capabilities and concluded that Platform-as-a-Service (PaaS) Cloud Computing platform would serve the purpose, PaaS also known as Application platforms in the cloud let developers rapidly implement and deploy Web-based applications usually emphasizes development productivity.

Keywords: Cloud Computing, Online Resources, PaaS, Application Platform.

I. INTRODUCTION

PaaS systems are Web-based platforms for developing applications. It provides end-to-end environments for online development of programs. Tasks such as deployment, editing code, debugging, management and runtime are handled. The provider of the system makes most of the choices that estimates how the infrastructure of the application operates, like the type of OS used, the programming language, the API's, and the capabilities of the management. Users build their applications with the on-demand tools and development environment. It provides a complete set of tools and technology, from the interface design, to integration, to process logic, to persistence. In some cases, developers can use a PaaS provider's online resources to build applications offline. Mostly, software developed with the technique runs online and is hosted by the PaaS service provider. The users can cache part of their program, work with it offline, and synchronize it later with the online application.

costs. PaaS creates a virtual platform for application development.

The user develops applications via a browser by interacting with the PaaS provider's servers. The system's programming model lets a developer specify business process flow and application logic without referring to the underlying physical computer systems and network interfaces. The model compiles program logic into the service calls used by the servers, client, and network interfaces. PaaS thus hides the complexity of the logic running between a client and a virtual server across a virtualized infrastructure. Some systems store the information that applications work within a user's infrastructure. Others store the information online, removing the need to call data from an external source and thereby reducing database- integration challenges. PaaS vendors provide various tools that offer different levels of precision for letting organizations track the usage of their PaaS-developed software.

They can track factors such as the number of times people using an application, the time they spend with a program, their activities with it, the software's performance etc. With some PaaS platforms, such as Bungee Connect, developers can build PaaS applications only when connected to the Internet. In others, such as the Google App Engine, the PaaS system creates an instance of the online server environment locally, allowing a developer to work offline and then push the application to the PaaS provider's server so that it can host the application. PaaS providers frequently try to make the programming experience familiar to developers by supporting common languages like C, Java, and PHP, or via drag-and-drop business logic tools that implement blocks of code and thereby reduce the amount of work required. However, providers such as

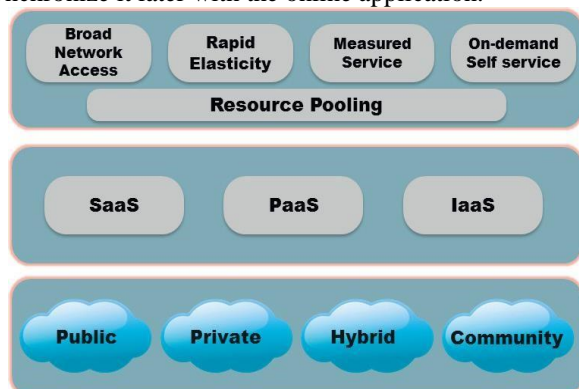


Fig. 1. Cloud computing model

The approach has numerous benefits such as increasing programmer productivity, enabling companies to build and release products more quickly, and reducing development

Bungee and Salesforce.com have developed their own languages. They say the new languages simplify the development process and are relatively easy to learn.

II. CHARACTERISTICS OF PAAS

[5] Five essential characteristics are enlisted for PaaS:

- **On-demand self-service:** The cloud user does not need any human interaction with the cloud provider.
- **Broad network access:** Cloud resources are available over the network and they are easily accessible.
- **Resource pooling:** Computing resources are shared among multiple users and assigned relative to the demand.
- **Rapid elasticity:** The provided resources scale rapidly (They appear to be unlimited.) according to the demand.
- **Measured service:** Resource usage is automatically monitored and optimized.

III. PAAS PROVIDERS

[1,2,4] Platform as a Service (PaaS) is a paradigm for delivering operating systems and associated services over the Internet without downloads or installations. An amount of leading companies are providing Cloud platforms, such as Microsoft, Google, Salesforce, Engine Yard and so on. Their platforms are briefly introduced in this section as well as their security procedures.

A. Azure

Azure Services Platform allows applications to be hosted and run at Microsoft data centres. It serves as a runtime environment for the applications and it also provides a set of services that allows customers to develop, manage and host their own applications offpremises. However, all Azure services and applications built on top of Windows Azure platform. Azure is running on Windows Server 2008 and has a web based configuration site which enable users to configure over the Internet. Moreover, servers stay behind firewalls to keep the service and data in a secure state. The platform also has load balancers which keep the servers from becoming hot spots in a highly automated virtualization environment. Again, everything belonged to .net security applies to Azure.

B. Bungee Labs

Bungee Labs' Bungee Connect targets small to medium-sized companies, as well as departments within larger organizations. It includes drag-and-drop user interface layout components and its own C-based language, called Bungee Logic.

The company's platform lets users deploy applications on servers managed by Bungee or on virtual servers running on Amazon's Elastic Compute Cloud.

EC2 provides a scalable Web services interface via which customers can request virtual machines on which they can load software. Organizations can also buy Bungee servers licensed to run within their own infrastructure, which gives them greater control and more internal auditing capabilities.

C. Cloud Foundry

Until recently, Cloud Foundry enjoyed a reputation as the most mature and widely used of available open source products. Not only is it supported by VMWare, but several other software vendors—Active- State being a notable example—have taken the basic Cloud Foundry platform and added much needed enterpriseready features, such as role-based security, chargeback and bill-back capabilities, and a full suite of cloud hypervisors. In December 2012, however, when VMWare spun off Cloud Foundry (along with its spring and gemfire divisions and EMC's Pivotal Labs and Greenplum) into a new company dubbed the Pivotal Initiative led by Paul Maritz, it was clear that Cloud Foundry had squandered its enterprise cachet and considerable market lead, having completely misunderstood how open source tools work. It is still an open question whether Pivotal Initiative can capitalize on this mashup of cloud development and big data tool synergies.

PaaS	Supported technologies	Development status	Open source	Application programming interface (API)	Supported databases
AppFog	Java, Python, Node, .Net, Ruby, PHP	Production	No	No	MySQL, MongoDB, PostgreSQL
AWS Elastic Beanstalk	Java, .Net, PHP	Production	No	REST API and CLI	MySQL
CloudBees	Java	Production	No	REST API and CLI	MySQL
Cloud Foundry	Java, Spring, Groovy on Grails, Ruby on Rails, Sinatra, NodeJS, .NET via extension	Beta	Yes	CLI	MongoDB, MySQL, and Redis
Cloudify	Java, .NET, Groovy, Ruby, C#, NodeJS, Spring, Chef	Production	Yes	CLI, REST API, Web	Cassandra, MongoDB, MySQL, HSQL
CumuLogic	Java, Spring	Beta	Yes	REST API	MySQL, MySQL cluster, MongoDB
dotCloud	Python, Ruby, NodeJS, Java, Perl, PHP	Production	No	REST API	MySQL, PostgreSQL, MongoDB, Redis
Google App Engine	Java, Python	Production	No	No	Google Cloud SQL
Heroku	Ruby, NodeJS, Clojure, Java, Python, Scala	Production	No	Yes	Amazon RDS, MySQL, MongoDB, Redis, CouchDB
Jetastic	Java, PHP	Production	No	Yes	Maxia DB, MySQL, PostgreSQL, MongoDB, CouchDB
Microsoft Windows Azure	C#, Java, PHP, Ruby	Production	No	REST API, storage	Azure SQL Database, CouchDB
RedPass	PHP, AS3, .NET	Production	No	Others storage/ CDN API	MySQL, 5 and MS SQL Server 2008
Red Hat OpenShift	Java, Java EE, Python, Perl, PHP, Ruby, NodeJS	Beta	Yes	REST API	MySQL, PostgreSQL, MongoDB
Salesforce Force.com	Apex	Production	No	REST API	Force Database
SAP HANA Cloud	Java, Java EE, JRuby, Python, Scala, JavaScript	Production	No	Web, CLI	SAP HANA, Sbase ASE, MaxDB

Table 1: Summary of Selected CloudPaaS Offerings

D. EngineYard

Engine Yard is a traditional hosting company in the Cloud Computing literature. The services which offered are called Clusters. Engine Yard attracts many customers because of its innovation in Ruby, Rails, which are new kids on the block in the web development area. That's to say it provides a platform for users to develop their own application based on Ruby, Rails and etc. Engine Yard uses professional security personnel to monitor and to physically control access to customers' data centers for 24/7. It has a core network infrastructure to segregate customer traffic to prevent eavesdropping and employs a variety of mechanisms and security checks to control data access.

E. Google App Engine

The platform offer scalable development and deployment environments. Developers also have access to Google's Big Table database, storage, and the same technologies for access control, security, and Web-services integration that the company's own applications use. The platform uses the popular Python scripting language. In addition, it includes an authentication system to validate the identity of programmers participating in the development process. Organizations can also utilize the system to authenticate the users of the applications they subsequently build. The

platform focuses on development of Web-based applications that require low-latency data storage and retrieval.

F. OpenShift

Red Hat's OpenShift, released in April 2013, is designed to appeal to a core enterprise customer base that not only finds open source tools attractive, but also values traditional vendor support. The environment looks promising, including a nice mix of rich development tools with some cloud and image management utilities wrapped around them.

The sophisticated, enterprise-ready platform should prove popular, and Red Hat's expertise in packaging open source products should keep Open-Shift on track despite its late market entry.

G. Salesforce.com

Salesforce.com's Force.com PaaS platform provides about 60,000 programs that users can tie together into bigger applications. They can also build their own applications from scratch. The platform provides a database for use with applications, numerous APIs, and logic capabilities, said Salesforce.com's Gross. Programmers can thus concentrate on other aspects of application development such as the user interface, he explained. Force.com uses a custom programming language called Apex, which focuses on database modeling. While applications must run on the Force.com platform, users can back up data offline.

IV. ADVANTAGES OF PAAS

[1] Each platform component is provided as a service. For ex: middleware –as-a -service , platform –as-a service, messaging –as-a -service, integration –as-a – service, communication-as-a-service

- 1) Provides services required to support the complete life cycle of building and delivering web applications and services on the internet.
- 2) Provides services to deploy, test, host and maintain application in the same integrated development environment (IDE).
- 3) Service provisioning includes multiple users concurrently using the same integrated application development environment.
- 4) Being a cloud offering, it follows cloud service model – pay per use / pay per go pricing model, where there is no need to buy the software, middleware or full year license; its pay on the basis of usage.
- 5) PaaS reduces TCO (Total Cost of Ownership), there is no need to buy all the system, software, platforms, tools and kits needed to build, run and deploy the application. Users can only rent them for the period for which services will be used. It changes the cost structure from Capital expense (Capex) to Opex (Operational expense) for an enterprise.
- 6) Built in scalability and elasticity to provide same efficiency and experience irrespective of load and usage.

- 7) PaaS is a perfect match for agile software development methodologies. An agile software development methodology is based on iterative and incremental development which may require iterations in need of software and other middleware platforms with progress and hence PaaS is the right match for agile application development methodology.
- 8) PaaS helps in the rapid construction of applications in the cloud by providing the necessary elements such as workflow amenities that are essential to the creation of a business application.
- 9) With PaaS, operating system features can be changed and upgraded frequently. Geographically distributed development teams can work together on software development projects.

V. DRAWBACKS OF PAAS

Some of the drawbacks of PaaS are as follows :

- 1) **Data security:** As with other cloud computing solutions, many companies still have low confidence in the level of data security offered by PaaS. Many businesses are still sceptical about having their applications hosted by a third party, while some enterprises and government clients need to be assured of compliance with all applicable regulations concerning security, privacy, and data retention before they decide in favour of PaaS services.
- 2) **Limited flexibility:** PaaS solutions can't match the flexibility of IaaS offerings. Unlike their IaaS counterparts, PaaS customers cannot necessarily create and delete multiple virtual machines easily. Again, when compared to SaaS offerings, PaaS falls short as it doesn't represent a complete product in the way that SaaS does. So, a company still needs to put in development effort to design, create, and test programs before they are deployed for the end users.
- 3) **Customer captivity:** With a limited number of PaaS vendors in the market today, each of which wants to build a binding relationship through a comprehensive offering, a vendor lock-in period is often the norm, which can limit the client's choices.
- 4) **Problems of integration with in-house systems and applications:** Integration of PaaS services with the rest of your systems and applications could often trigger an increased complexity.

V. CONCLUSION

The findings we present here could be useful for both developers and platform vendors. The framework creates awareness for what to look for in the first place, and collects the information necessary for an informed decision. The identified technical and nontechnical design choices provide further insights into whether to select a particular product. They let developers choose or avoid solutions that implement a design choice that's beneficial or hindering for their productivity. Vendors, on the other hand, receive first-hand insights into how developers perceive their design choices, and can use this information to improve their products.

REFERENCES

- [1]. George Lawton, Developing Software Online with Platform-as-a-Service Technology,IEEE,2014.
- [2]. Minqi Zhou, Rong Zhang, Dadan Zeng, Weining Qian, Services in the Cloud Computing Era: A Survey, National Institute of Information and Communications Technology,IEEE,2014.
- [3]. Mehmet Tahir Sandikkaya, Ali EmreHarmancı, Security Problems of Platform-as-a-Service (PaaS) Clouds and Practical Solutions to the Problems,IEEE, 2014.
- [4]. Beth Cohen, PaaS: New Opportunities for Cloud Application Development,IEEE,2014.
- [5]. Oliver Gass, Hendrik Meth, and Alexander Maedche, PaaS Characteristics for Productive Software Development ,IEEE, 2014.
- [6]. Xinlei Wang, Yuba Tan, Application of Cloud Computing in the Health Information System, IEEE, 2010.
- [7]. Jannis Rake-Revelant, Oliver Holschke,Platform-As-A-Service For Business Customers, IEEE, 2010.
- [8].