Cloud Computing Solutions for Better Websites and Applications

Prepared for:

Renee Deveaux
Communication Department Professor
Columbus State Community College

Prepared by:

Brandon Sersion
Columbus OH, 43302
Software Development Major
Columbus State Community College

Course:

COMM 2204

July 19, 2017

110 Angelbrook Avenue
Columbus, OH 43204
July 19, 2017

CSCC Communication Department Professor:
Renee Deveaux
550 East Spring Street
Columbus, OH 43215

Dear Professor Deveaux:

Enclosed is my report on cloud solutions for website architectures, and how they can save time and money over the traditional server paradigm. The reason I prepared this report was to help decide which cloud architecture I should use for the website I am currently working on.

The project was a success. Not only am I now equipped to make that decision, I realized I can use multiple different architectures for different features, picking and choosing which architecture is best for which purpose. I now have a good idea of how to architect every feature on my website. After a year of learning and preparation, I am finally ready to start production.

Sincerely,
Brandon Sersion

Brandon Sersion
513-488-3654

Table of Contents

List of Illustrations

Executive Summary

The traditional server infrastructure for websites and applications has gone obsolete. Cloud Solutions are much cheaper, simpler, and more popular. But which of the four popular cloud architectures are the best? Infrastructure as a Service, Platform as a Service, Container as a Service, and Software as a Service all have their pros and cons. Whichever one is “the best” depends on the situation.

In short, IaaS is open-ended and difficult, PaaS is “on rails” and easy, CaaS is hybrid but buggy, and SaaS is zero effort but expensive.

Introduction

More and more websites are outsourcing their back-end compute workload and databases to “as a service” cloud solutions, instead of the traditional “on-premises” method of renting or owning a server outright:

1. Servers have far more compute power than most users would ever need. With a cloud solution, the user can pay for a smaller amount.
2. Simply renting a server costs on average $1476.31 a month. Renting the same amount of compute power through an IaaS cloud vendor (a virtual identical offering from the user standpoint) costs about $313.90 a month. Additionally, many vendors offer generous free trials. [Boisvert]
3. Servers have to be set up and maintained meticulously. This is a bad idea unless the company has a server expert.
4. Most “as a service” vendors guarantee 99.99% uptime. It is not a good idea to rely on servers that have sporadic Internet access or power.
5. Cloud infrastructure services have mature software that is used to reliably manage compute processes from anywhere in the world.

Subject & Purpose

The subject of this report is the four different “as a service” cloud computing options – Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Container as a Service (CaaS), and Software as a Service (SaaS). The purpose of this report is to help application programmers or web developers decide what cloud computing option to use for which processes in their larger application.

Scope

The scope of this project is “as a service” cloud computing architectures – to explain what they are and their benefits and disadvantages. Benefits and disadvantages considered include:

1. Can the architecture do what I need it to well?
2. Ease of learning
3. Ease of development
4. Ease of administration
5. Vendor fees
6. Vendor lock-in
7. Popularity (what are the chances development ceases?)
8. Open source

[Feinberg]

Limitations

First, this report will focus on “as a service” architectures themselves, not the plethora of different companies that offer each one. This report does however use specific examples of companies to describe how the architectures work. Second, this report will not talk about traditional web hosts. Third, it is difficult to see past marketing hype in this industry. Companies tend to hide what their service cannot do. This report attempts to see past this.

Methods of Collection

- Company promotional and billing information
- Web journals on cloud computing
- Recordings from seminars posted to Youtube

Plan of Organization

The organizational plan for this research project was:

- First, to explain the difference between traditional on-premises server architectures and cloud computing architectures, and why many companies are moving over.
- Second, to introduce the features companies are looking for in their on-premises or cloud applications.
- Third, to explain the four main cloud computing architectures, and how well they do in the features department versus the others.

Definitions & Acronyms

ACID (Atomicity, Consistency, Isolation, Durability)
Refers to SQL databases that have all these features, therefore are good for high demand database transactions, for example financial transactions.

CaaS (Container as a Service)

IaaS (Infrastructure as a Service)

PaaS (Platform as a Service)

SaaS (Software as a Service)

On-premises
Refers to a piece of software run directly on the user's hardware, owned or rented.

API (Application Programming Interface)

CPU (Central Processing Unit)

GPU (Graphics Processing Unit)

Infrastructure as a Service (IaaS)

Infrastructure as a Service provides virtual machines with predefined compute resources. Virtual machines are controlled and operated just like a personal computer at home, through a hypervisor software application. The user has complete control over the system, to install any operating system or software. To illustrate this point, security experts tested the recent WannaCry virus using virtual machines. Although the virus wreaks havoc on said virtual machines, the virus is of no risk to other

virtual machines on the same server because virtual machines are self-contained. When the user is done with a virtual machine, they can shut it down and start a clean one, no extra charge. IaaS typically offers an assortment of virtual machine templates, optimized for various workloads. For example, Microsoft Azure Virtual Machine offers:

- General Purpose – Balanced CPU-to-memory ratio. Designed for testing, development, small to medium databases, and low to medium traffic web servers.
- Compute Optimized – High CPU-to-memory ratio. Designed for medium traffic web servers, network appliances, batch processes, and application servers.
- Memory Optimized – High memory-to-CPU ratio. Designed for relational database servers, medium to large caches, and in-memory analytics.
- Storage Optimized – High disk input and output. Designed for Big Data, SQL, and NoSQL databases.
- GPU – Specialty virtual machine with one or multiple graphics processing units – designed for heavy graphic/video rendering.
- High Performance – Specialty virtual machine with super heavy stats.

All Azure virtual machine varieties come in a selection of sizes. Load-balancing between virtual machines is free. Outgoing data bandwith is charged separately. [“Linux Virtual Machines Pricing.”]

IaaS is the cheapest of the four architectures, but sometimes IaaS requires hard work to realize that discount. It is always cheaper to scale out (more virtual machine instances) than to scale up (larger virtual machine instances). The discount from scaling out gets exponentially bigger as the virtual machine gets larger. However, scaling out requires more complex configuring than if the user contains the entire process to a single large virtual machine. Scaling out requires the user to develop a system for load balancing. Databases are downright impossible to scale out and retain strict ACID requirements – in order to achieve partition tolerance, the user is forced to sacrifice either consistency or availability. [Messinger.] IaaS is indisputably the cheapest option for a process the user can easily contain to a single small virtual machine. IaaS may not be the cheapest if it requires a significant time investment figuring out how to scale out a large process – it depends most of all on the skillset of the team.

IaaS has several advantages:

1. Complete user freedom
2. Cheapest generally
3. Many bill by the minute – great to power up a machine briefly for testing purposes
4. No proprietary APIs causing vendor lock in. It is relatively simple to migrate from one IaaS vendor to another

IaaS has several disadvantages:

1. Sacrifices ease of use
2. Sometimes the user is forced to pay for extra resources they do not need. This can be significant if the actual usage is in between two disparate pricing tiers
3. Complex software administration – the user handles updates including vital security updates that can temporarily take the virtual machine temporarily out of service

Platform as a Service (PaaS)

Platform as a Service provides raw compute power – the PaaS vendor offers a software interface the user can use to upload their program with a click of a button, and manage it with a simple user interface. No provisioning of server space is needed. PaaS is perfect for non-expert programmers who want to get a website or application up and running fast. On the downside, the user has no other choice but to use the user interface provided. If the PaaS vendor does not explicitly offer a feature, the user cannot change their servers to start supporting it. Most configuration is controlled exclusively by the vendor.

PaaS charges based on resources used, like an electric bill. PaaS is much more expensive than IaaS or CaaS, but less expensive than SaaS.

PaaS has several advantages:

1. Easy to use, pre-defined software modules
2. Pay for what is needed
3. Easy software administration – the vendor handles things like auto-scaling and load balancing.
4. The vendor handles updates, including vital security updates – the process never has to go offline.

PaaS has several disadvantages:

1. Can use only pre-defined software modules – usage limited by what the vendor offers
2. Expensive
1. Security is only as good as the vendor's security
2. Proprietary APIs cause vendor lock in

Containers and Container as a Service (CaaS)

Container as a Service is basically a hybrid between IaaS and PaaS. A container is a self-contained software package which contains everything it needs to run – code, dependencies, settings, and host operating system. Multiple containers can share the same operating system kernel, reducing application compute and memory usage compared to IaaS. Containers also make deploying applications simpler than with IaaS, because everything is in one place – fixing the common issue where an application runs differently, or refuses to run, on different computers The self-contained nature of containers makes it easy to migrate them from one vendor to another, reducing vendor lock-in as compared to PaaS. Since the software in a container is completely client-side, users have pretty good control of what they can do with them, as compared to PaaS. ["What Is a Container."]

Containers have some drawbacks. One, they are a very new technology so the user may have to develop even basic functions on their own. Second, the fact that different containers share a kernel creates a security vulnerability. The user definitely would not want to download the WannaCry virus onto a container, it might escape. Third, containers are not permanent like IaaS virtual machines. The second the container turns off, data is lost. Containers can be problematic for databases. Containers also have a heavy learning curve.

Containers can be used in IaaS virtual machines, but CaaS specifically refers to a subset of PaaS services that allow the user to upload containers instead of lines of code. CaaS charges based on the

resources used and skirts vendor lock in. The user can move a container from an IaaS to a CaaS and vice versa at any time, very little hassle.

Containers have several advantages:

1. Performance of a PaaS
2. Freedom and low cost of an IaaS
3. Mitigates Vendor Lock In – can even move a containerized application from an IaaS virtual machine to a CaaS (technically a variation of PaaS)

Containers have several disadvantages:

1. New technology, the user might run into bugs or have to develop essential features on their own
2. Security issues
3. Non-permanent – lose data when turned off. Not ideal for databases.

Software as a Service (SaaS)

Software as a Service offers user-facing software applications run on the vendor's servers, which the user can log in to using an Internet browser. SaaS is different than the three other cloud computing architectures in that each SaaS is designed for one specific purpose. Many companies offer SaaS versions of their existing popular desktop applications. For example, Adobe offers SaaS versions of Photoshop, Illustrator, InDesign, Lightroom, Adobe Premiere Pro, and Experience Design. ["Adobe Creative Cloud"] Microsoft office offers SaaS versions of Word, Excel, Powerpoint, Outlook, OneNote, OneDrive, Publisher, and Access. ["Welcome to Office."] Other examples of Software as a Service include widgets embedded into a website to do a certain job. For example, many websites use Google's log-in SaaS instead of managing their own in-house username and password database.

SaaS is the most expensive of the four architectures, by far. They typically charge a hefty fee per user per month. Adobe Creative Cloud, for example, costs $49.99 per user per month. On the plus side, a user is allowed to access that SaaS from many different computers. If the user has 50 different computers they need to run the program on different days, it may actually save a significant amount of money (over purchasing the program outright 50 times.) SaaS offerings also excel at multitasking. Many are designed so that two people can work on one file at the same time.

SaaS has several advantages:

1. Typically requires zero programming ability
2. Typically the most polished software on the market because the company specializes in perfecting one very specific thing
3. Excels at difficult to implement features, like multiple people being able to synchronously edit a single document

SaaS has several disadvantages:

1. Each service has an extremely limited functionality
2. No way to expand that functionality
3. Most expensive. The price especially adds up when using multiple different services

Conclusion

The primary difference between on-premises applications and the four cloud architectures are user control versus vendor control.

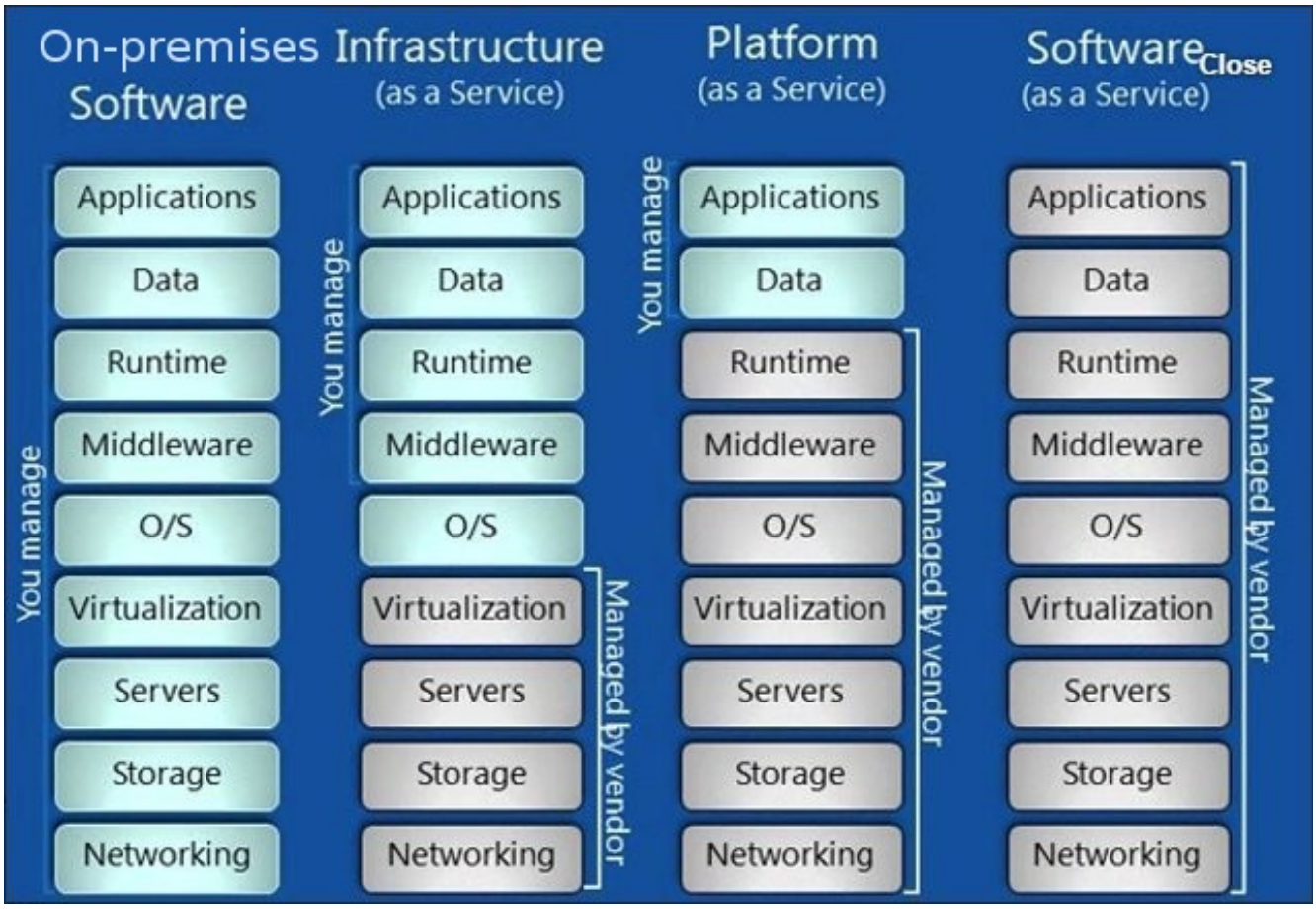

Figure 1 – Architectures compared by user versus vendor control. ["Private vs. Public vs. Hybrid Cloud: Which One to Choose?"]

The most difficult and most time consuming solutions are the cheapest, but also have the most user freedom and avoid vendor lock-in.

Recommendations

Overview

	Vendor Fees	User Control	Ease of Learning	Ease of Development	Ease of Administration	Vendor Lock-in
CaaS	5	5	1	3	1	No
IaaS	5	5	2	1	2	No
PaaS	3	3	4	4	4	Compatibility layers can mitigate
SaaS	1	1	5	5	5	Unavoidable

Figure 2 – Architectures rated by features. 1 is worst, 5 is best.

Ideal for

IaaS: People wishing to maximize the efficiency of their programs and cash savings, at the expense of ease of use – large companies, tech experts.

PaaS: People wishing to get a program up and running quickly and with little fuss – small businesses, tech startups, people new to website or application development.

CaaS: People willing to take a risk with a new technology and perhaps reap the benefits of both PaaS and IaaS, at the expense of possible technical woes and bugs – small businesses, tech startups, large corporations.

SaaS: People hoping to outsource a program's management and operations so they do not have to manage it – large corporations, medium businesses.

Personal Website

According to businessman Bob Bender, ease of use is extremely important with the technology a small business uses. The more complex something is, the more time consuming it will be, and the greater the likelihood there is for technical problems that freeze company operations. On his advice, Brandon's website will use PaaS for most of its architecture, and outsource some of the basic functions like the website's login system to SaaS providers, simpling the back-end even further. Brandon might try CaaS, once the technology becomes more mature, but right now the possibility of bugs is too big a problem. IaaS is too complex for the most part, but will be used for testing functions before they are officially released.

Works Cited

"Adobe Creative Cloud." *Adobe*. N.p., n.d. Web. 19 July 2017. <https://www.adobe.com/creativecloud.html>.

(B. Bender, personal interview, June 28, 2017).

Boisvert, Guillaume. "Cost of Server Ownership: On-Premise vs IaaS." *SherWeb*. N.p., 21 Sept. 2015. Web. 19 July 2017. <http://www.sherweb.com/blog/total-cost-of-ownership-of-servers-iaas-vs-on-premise/>.

Colman, Eamonn. "When to Use SaaS, PaaS, and IaaS." *ComputeNext*. N.p., 27 Aug. 2013. Web. 19 July 2017. <https://www.computenext.com/blog/when-to-use-saas-paas-and-iaas/>.

Feinberg, Leonid. "Cloud Technology 101: How to Choose the Right Cloud Infrastructure for Your Company." *CloudEndure*. N.p., 12 June 2017. Web. 18 June 2017. <https://www.cloudendure.com/blog/cloud-technology-iaas-paas-saas/>.

Kehoe, Ben. "Serverless vs. PaaS and Docker: Three-legged Hairless Yaks." *Serverless.Zone*. N.p., 17 Nov. 2016. Web. 18 June 2017. <https://serverless.zone/serverless-vs-paas-and-docker-three-legged-hairless-yaks-f2c4191b1ed6>.

"Linux Virtual Machines Pricing." *Pricing - Linux Virtual Machines*. Microsoft Azure, n.d. Web. 19 July 2017. <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>.

Messinger, Lior. "Better Explaining the CAP Theorem." *Dzone.com*. DZone, 17 Feb. 2013. Web. 19 July 2017. <https://dzone.com/articles/better-explaining-cap-theorem>.

"Private vs. Public vs. Hybrid Cloud: Which One to Choose?" *Logicworks*. N.p., 24 Mar. 2015. Web. 18 June 2017. <http://www.logicworks.com/blog/2015/03/difference-private-public-hybrid-cloud-comparison/>.

"Welcome to Office." *Office*. Microsoft, n.d. Web. 19 July 2017. <https://www.office.com/>.

"What Is a Container." *Docker*. N.p., n.d. Web. 19 July 2017. <https://www.docker.com/what-container>.

Williams, Adam. "Knowing Your aaS from a Hole in the Ground: Exploring SaaS, PaaS, CaaS, and IaaS." *YouTube*. KnoxPHP, 08 May 2017. Web. 18 June 2017. <https://www.youtube.com/watch?v=vXoz5Yg8KtM>.