Cork Institute of Technology

M.Sc. Cloud Computing

Productive Clouds

Scale-out the base or scale-up the height?

Richard Scallan

R00143768

Supervisor: Ted Scully

Richard Scallan

08 May 2017

Productive Clouds

Scale-out the base or scale-up the height?

Abstract

This paper contributes to research on the factors influencing the adoption of cloud technologies and the influence that benchmarking of performance and cost has on adoption. Within the paper is substantial research providing a significant comparative analysis of the performance and cost efficiencies of scale-up and scale-out deployments.

A review of the current status of cloud computing is detailed. Additionally the overall adoption of the various cloud models (IaaS, PaaS, SaaS) is highlighted and research into the factors influencing cloud adoption levels is provided. Items which are factors in the decision to adopt cloud such as, cost of adoption, performance, scalability and security are reviewed as part of this. The comparison between scale-up and scale-out cloud deployments is considered and empirical research is completed against both configurations. A contribution of this empirical research was the use of an application designed for both scale-up and scale-out deployments (MongoDB) deployed on Public Cloud technology (Amazon Web Services EC2) and benchmarked with a single tool (YCSB). The goal of the research is to investigate and understand if the some of the primary reasons for a delay in cloud adoption (performance, cost efficiency) is influenced by the configuration method.

The results demonstrate that should the applications requirements be accurately identified prior to deployment, that a scale-up strategy can deliver better cost efficiencies. The efficiencies are relative to the performance and latencies required by the application. The granular scalability and predictable performance growth provided by a scale-out deployment is realised in the resulting metrics. Scaling-out provided for a reduced latency while not being quite as efficient with resources. Additional contributions provided include the performance measurement when the number of threads used by the benchmark application are also evaluated. The number of threads is shown to have a significant impact on the overall efficiency and productivity metrics observed in all benchmark runs and had not been recorded previously. The purpose of the benchmarking and research being to providing accurate metrics gathered across the same platform to enable an informed comparative conclusion to be made.

Acknowledgements

I would like to thank Dr Ted Scully for his guidance and supervision on this research project. I would like to thank my family and friends, for their help and support while I have been completing this project. I would like to especially thank my wife Sarah and our children Cillian and Dara as they have provided understanding and patience during the time I needed to complete this assignment.

Vocabulary

Phrase / Acronym	Meaning
NIST	National Institute of Standards and Technology
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
Opex	Operating Expenditure
Capex	Capital Expenditure
IoT	Internet of Things
CSP	Cloud Service Provider
AWS	Amazon Web Services
EC2	Elastic Cloud Compute
ECU	EC2 Compute Unit
YCSB	Yahoo! Cloud Serving Benchmark
QoS	Quality of Service

Table of Contents

1 Introduction

In the last decade the adoption of cloud computing has been significant. Most companies leverage cloud services as a growing part of their IT strategy. 95% of organisations surveyed in the 2017 "State of the Cloud" report responded that they were either currently running applications in the cloud or experimenting with Infrastructure as a Service (IaaS) (RightScale, 2017). Enterprises are therefore leveraging a hybrid approach, leveraging cloud technologies for particular use cases while keeping some workloads onsite for a variety of reasons.

This paper provides a review of the current state of cloud computing usage in enterprises. There is a summary of the background research conducted into cloud computing and the different characteristics (self-service, resource pooling, elasticity, etc.), deployment (Private, Public, Community & Hybrid) and service (IaaS, PaaS & SaaS) models of which it consists. A review of the benefits of cloud computing and the current adoption rates across enterprises is also provided. Many of the benefits, such as cost, scalability, security, performance, are often touted and blindly associated with cloud computing. They are however, less often reviewed and investigated as items which may negatively influence cloud adoption. As cloud adoption continues to grow, development in new technology is directed further away from traditional onsite IT environments, to the more consumable cloud environments. An investigation into the factors which influence cloud adoption is required and performed within this paper. Reviewing the factors influencing adoption, positively and negatively, provided an understanding as to what areas to investigate further and gather metrics against.

Following detailed reviews of the key influencing factors, it could be concluded that as scalability and efficient resource provisioning are key benefits, so too should the performance and cost of cloud computing be positively viewed. As this is not always the case and as is shown, the performance and cost of cloud computing can be negative influences on cloud adoption. As scalability and access to unlimited resources is a key benefit of cloud, then why can performance not scale as required? Similarly for the cost, as cloud provides the ability to provision the exact resources required, adding and releasing as needed, why is the cost of adoption often a negative influence? One of the objectives of this paper was therefore to review the reasons for these negative influence. The research showed that overall performance and efficiency of cloud are those which appear to have the greatest impact while simultaneously be items which are more measurable. Other aspects such as security and portability are also extremely important but are less easy to quantify.

Reviewing the reasons for the contradiction in performance and cost being both benefits of and factors affecting adoption of cloud pointed to number of things. The primary being a lack of clarity around the overall cost of adoption and running workloads with particular performance requirements in the cloud. An evaluation of causes for this lack of clarity indicated that it stemmed from inconclusive and vague cloud benchmarking findings. Following an analysis of various benchmarking tools capabilities, it was observed that cloud performance expectations were typically hard, but not impossible, to determine from tools not specifically designed for cloud. This lack of definitive benchmark metrics meant that cloud adopters typically either over

or under provisioned their cloud deployment as they could not be certain of the actual requirements. This over or under provisioning then negatively impacted the overall performance or cost and efficiency of various cloud deployments and consumers cloud experience. The development of more appropriate benchmark tools has provided for a more informed decision to be made. A review of the research done previously into benchmarking of cloud performance and cost led to the recognition a further area of interest to which significant contribution through this paper could be made. That is the comparison of public cloud scale-up vs scale-out deployments when considering the performance and cost of running workloads.

There is almost an expectation that scale-out is always better. For this reason the research purpose is also to challenge the perception that cloud configurations are best deployed in one particular way, typically considered as scale-out. The paper investigates the effectiveness of scale-up and scale-out across a number of metrics to provide sufficient detail to form an educated assessment. It builds on previous recent and relevant research and contributes to the overall understanding of the factors influencing cloud adoption and the impact of using scale-up or scale-out deployments (Hwang, et al., 2016). With this in mind the primary typical benefits of scale-out deployments, such as ease of scalability and high availability, should be set aside. The performance, efficiency and productivity of a scale-up versus a scale-out configuration needs to be considered in line with their relevance to influencing adoption of cloud.

As highlighted, the primary objective behind this research project is to review the ongoing adoption of cloud computing and investigate the factors which can potentially influence the rate of adoption. As outlined, the research done means that there were subsequent objectives identified which required additional exploration. These included the need to evaluate the current performance and cost metrics used by potential cloud consumers when considering cloud adoption. It is necessary to determine if these metrics provide a sufficient base for decisions or if further items should be considered. Amazon Web Services (AWS) Elastic Compute Cloud (EC2) was determined as the appropriate public cloud service to use as the basis for the research and both scale-up and scale-out deployments.

The effect of particular benchmarking tools and applications is a necessary consideration of this paper also. The purpose of reviewing the benchmarking tools and configuration is to ensure that influencing factors on cloud adoption such as cost efficiency and performance are accurately measured. For this reason the research will identify and analyse the options to be considered when gathering the metrics. The benchmark tool deployed and used for this research was YCSB (Yahoo! Cloud Serving Benchmark) and the application used was the NoSQL database MongoDB. The subsequent key contribution of this research is a comparative analysis of the results seen when benchmarking scale-up and scale-out cloud deployments from both a performance and cost efficiency perspective. This provides for empirical results to be delivered to ensure that both scale-up and scale-out deployments can be accurately compared and the factors influencing adoption of cloud reflected upon.

The paper is constructed in the following manner. The initial background research on Cloud Computing and its associated benefits and factors influencing adoption was carried out and documented in Chapter 2. Within Chapter 3 the empirical research methodology is described in detail. It includes information on the main technologies used such as AWS, MongoDB and YCSB. Chapter 4 details the empirical results and is split into two sections. The first outlines the scale-up deployment and provides both documentation and graphs of the subsequent results and the second section provides the same detail for the scale-out deployment. Chapter 5 outlines the conclusions of the paper and includes suggestions for areas of further research which could be considered to build upon the findings outlined in this thesis.

2 Background Research

2.1 Cloud Computing Definition

Cloud computing can be defined as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."* (Mell & Grance, 2011)

This definition was created by the National Institute of Standards and Technology (NIST) and agreed upon by industry and governments alike. Cloud computing provides the first significant change in the method of delivering, managing and consuming IT services. The increase in connectivity between the global community, business and personal, has meant that the requirements and demands for more compute, network and storage capabilities has likewise rapidly increased. The capabilities and accessed by consumers have driven IT to provide a more modern datacentre. The requirement for IT to be "consumed" has led to the requirements of these modern data centres to become more responsive and capable to provide services in response to changes in demand. This "consumerization" (Gartner, 2017) of IT has encouraged the adoption of cloud technologies due to the capabilities it can provide. As according to NIST, cloud computing has 5 Essential Characteristics, 3 Service Models and 4 Deployment Models as outlined in figure 1 below and to be discussed in the following sections.

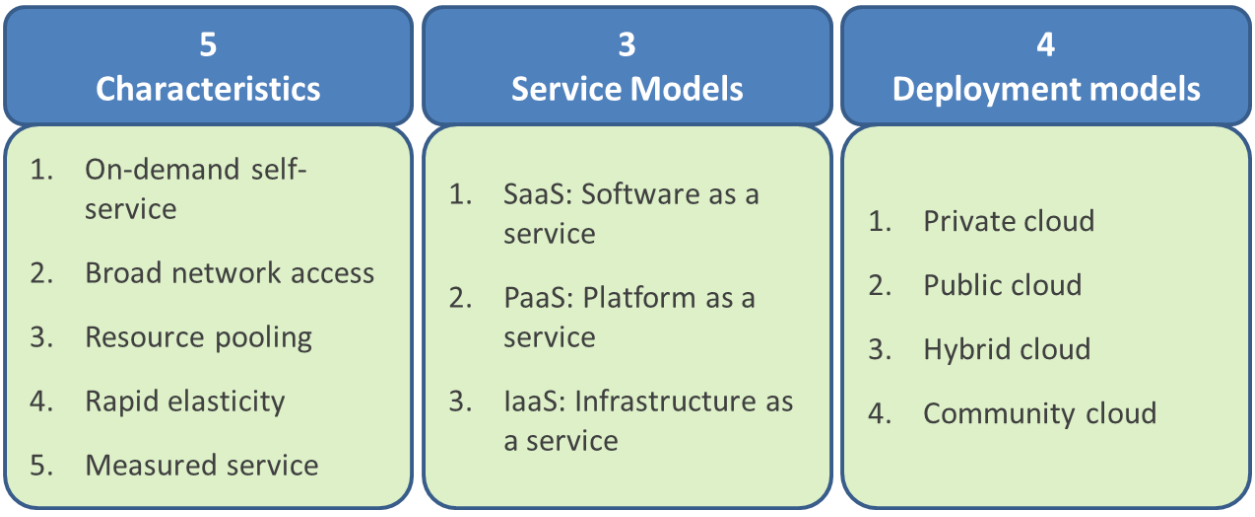

Figure 1 - NIST definition of Cloud Computing

2.1.1 Essential Characteristics of Cloud

NIST identifies the following essential characteristics of the technologies being used in the modern data centre which provide the capabilities to deliver cloud services. These characteristics are shown in figure 2 below and more details follow.

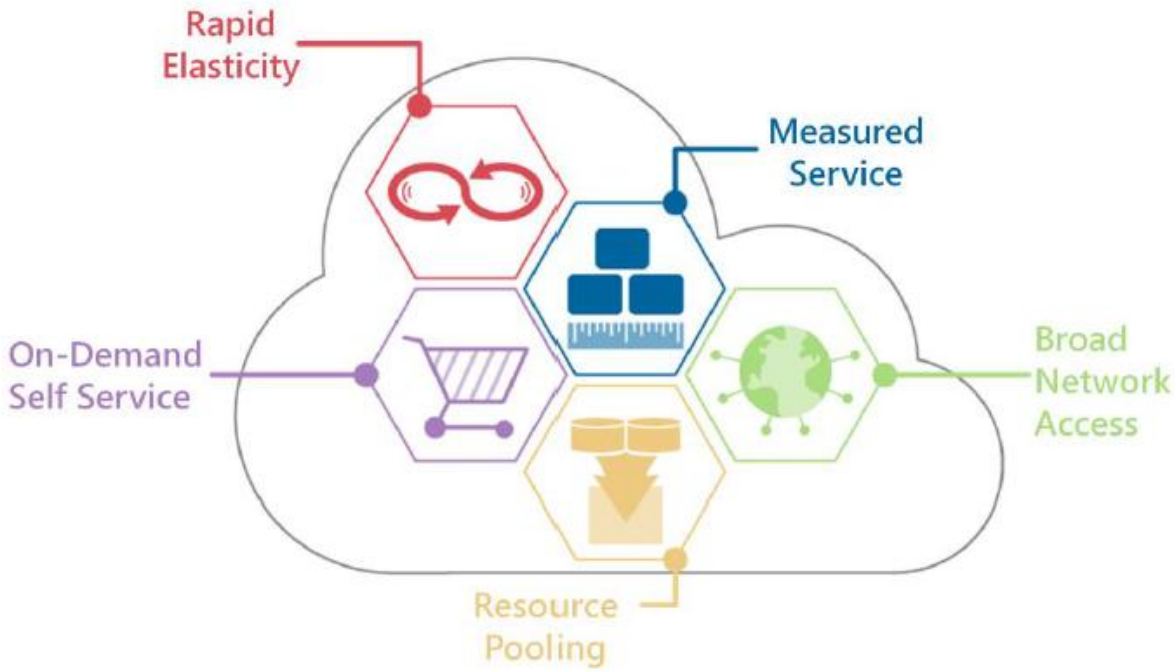

Figure 2: 5 Essential Characteristics of Cloud Computing (NIST) - Image from Dell EMC

- **On-Demand self-service**

  "A consumer can unilaterally provision computing capabilities, such as server time or networked storage, as needed automatically without requiring human interaction with each service provider."- (Mell & Grance, 2011)

  In cloud computing the consumers have the ability to provision whatever IT resources they need from the cloud. The resource should be available for provisioning on demand, at any time. Self-service meaning that the consumer provisions the resource by carrying out all the necessary activities themselves.

- **Broad network access**

  "Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g. mobile phones, tablets, laptops, and workstations)." - (Mell & Grance, 2011).

  The cloud service should be accessible to consumers on any end-point device from anywhere. Whether they are on the internet or an enterprises private network, users should be able to access the service without the need for specialised client software on the device. The network communications should be based on any which use the standard network specifications, protocols etc. detailed in the OSI (Open Systems Interconnection) model and the TCP/IP protocol suite. Additionally HTTP and other standard web services can be used to enable the applications to communicate.

- **Resource pooling**

"The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence. In that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth." - (Mell & Grance, 2011).

Resources such as compute, memory, storage and network are combined into one large pool in cloud computing. This enables the Cloud Service Provider to share this pool and serve multiple consumers. It also allows the IT resources to be automatically and dynamically assigned, released, and reassigned according to the requirements and demand of consumers. The benefit of this resource pooling is that CSPs can flexibly provision and reclaim resources from an environment with a high level of utilization. Consumers have a greater degree of flexibility than in a traditional environment as they can provision and release resources from / to the pool when required. This pooling means that there are typically multiple consumers serviced through this single set of resources. Resource pooling and sharing of the cloud resources lower the cost of services for consumers as they only pay for the resources that they use. Even though the underlying resources are shared, the consumers are logically separated. The customer typically has no information or control over the specific location of the resources but could provide requirements at a higher datacentre / country level.

- **Rapid elasticity**

  "Capabilities can be rapidly and elastically provisioned, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time." - (Mell & Grance, 2011).

  Rapid elasticity refers to the ability for consumers easily provision or release additional resources either manually or automatically on demand. This characteristic provides consumers with the sense that there is unlimited IT resources available to be provisioned from at any time and allows them to adapt quickly to varying workloads requirements. Resources can be quickly and dynamically (through the use of manual or automatic monitoring and provisioning) expanded (scale-out) or reduced (scale-in) in order to maintain the performance or service level required.

- **Measured service**

  "Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service."- (Mell & Grance, 2011).

Cloud systems automatically provide a metering capability that generates bills for the consumers based on the usage of services. It continuously monitors resource usage per consumer, and provides details on utilization of all consumable resources such as processor time, network bandwidth, and storage capacity. Metrics related to the current utilization of the cloud enable cloud providers to accurately plan and deliver for changes in capacity and performance and optimise the cloud service's usage.

2.1.2 Cloud Deployment Models

NIST created formal definitions of the 4 cloud deployment models which are summarised in figure 3 and the descriptions below.

Figure 3: 4 Cloud Deployment Models (NIST) - Image from Dell EMC

- **Private cloud**

  The cloud infrastructure is provisioned for exclusive use by a single organization. It may be owned, managed, and operated by the organization or a third party, on or off premises. The main criteria defining Private Cloud is that the capabilities are provided for the exclusive use of the organisation. It may support different tenants in the form of business units but is not shared with any other organisation.

- **Public cloud**

  The cloud infrastructure is provisioned for open use by the general public. It is typically owned, managed, and operated by a government, academic or business organisation. The infrastructure is hosted on the premises of the cloud provider but can be distributed to multiple tenants through logical

separations. Consumers of Public Cloud services range from a single user to large enterprises. Some of the well-known providers are Amazon Web Services, Microsoft, Google and Rackspace.

- **Community cloud**

  The cloud infrastructure is provisioned for exclusive use by a specific community of consumers. These consumers are typically organizations that have shared concerns or specific requirements such as security or compliancy. One such example would be a cloud provided for, managed and operated, and consumed by a specific group of healthcare organizations designed to deliver services meeting their specific requirements.

- **Hybrid cloud**

  The hybrid cloud infrastructure is a composition of two or more of the previously defined cloud infrastructures (private, community, or public). In hybrid cloud, the composite cloud platforms remain separated and unique. They are however bound together through standard or proprietary technology which enables data and application portability.

2.1.3 Cloud Service Models

There are three main types of cloud services that consumers can provision. NIST refers to these as Cloud Service Models. Shown in figure 4 below are the three service models and highlighted is the amount of the IT stack which is managed by each of either the cloud consumer or cloud provider.

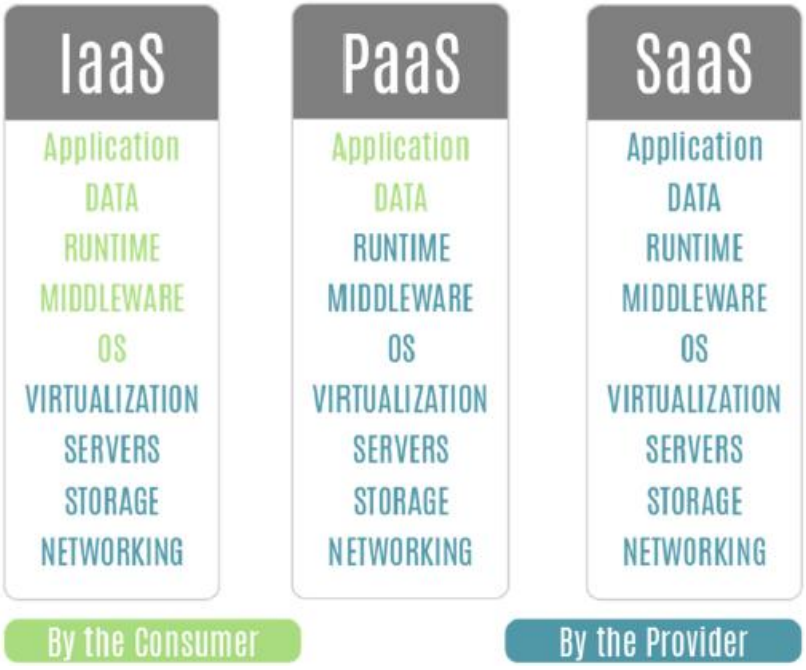

Figure 4: Three Cloud Service Models (NIST) - Image from Dell EMC

- **Infrastructure as a Service (IaaS)**

  The consumer has the ability to provision processing, storage, networks, and other fundamental computing resources. They would have control of the operating system and applications that are deployed and possibly limited control of select networking components (e.g. host firewalls). While consumers have access to these resources, it is the provider who maintains control of the infrastructure beneath as is seen in figure 4 above. An example of IaaS would be Amazon Web Services (AWS) and their EC2 service.

- **Platform as a Service (PaaS)**

  Platform as a Service is the method of providing consumers with the capability to deploy and control applications which are created using programming languages, tools, libraries, and other services supported by the provider and deployed onto cloud infrastructure. Consumers have control over the deployed applications and configurations whereas the PaaS providers maintain control over the fundamental infrastructure and operation systems. Pivotal is the best known provider of PaaS currently.

- **Software as a Service (SaaS)**

  An application deployed on cloud infrastructure which is controlled by the provider. The consumer has the ability to use and potentially change their settings on the cloud provider's applications running on a cloud infrastructure without being able to manage or control the underlying cloud infrastructure. Some examples of Software as a Service would be Salesforce.com or Gmail.

2.2 Benefits of using Cloud Computing

The main benefits of using Cloud computing technologies, be they on or off-premise IaaS, PaaS or SaaS, are discussed in many publications. The top reasons for using or considering to use public cloud services stated by Gartner survey respondents are shown in figure 5 below. The following provides a comprehensive overview of these benefits derived from an extensive review of literature. More specifically this section focuses on the following benefits of adopting cloud-based technologies: (i) reduced cost, (ii) faster time to value, (iii) flexibility, (iv) scalability, (v) innovation, (vi) high availability & business continuity and finally (vii) security (Gartner, 2016) (Rittinghouse & Ransome, 2009) (Marston, et al., 2011) (Carroll, et al., 2011) (Bhardwaj, et al., 2010).

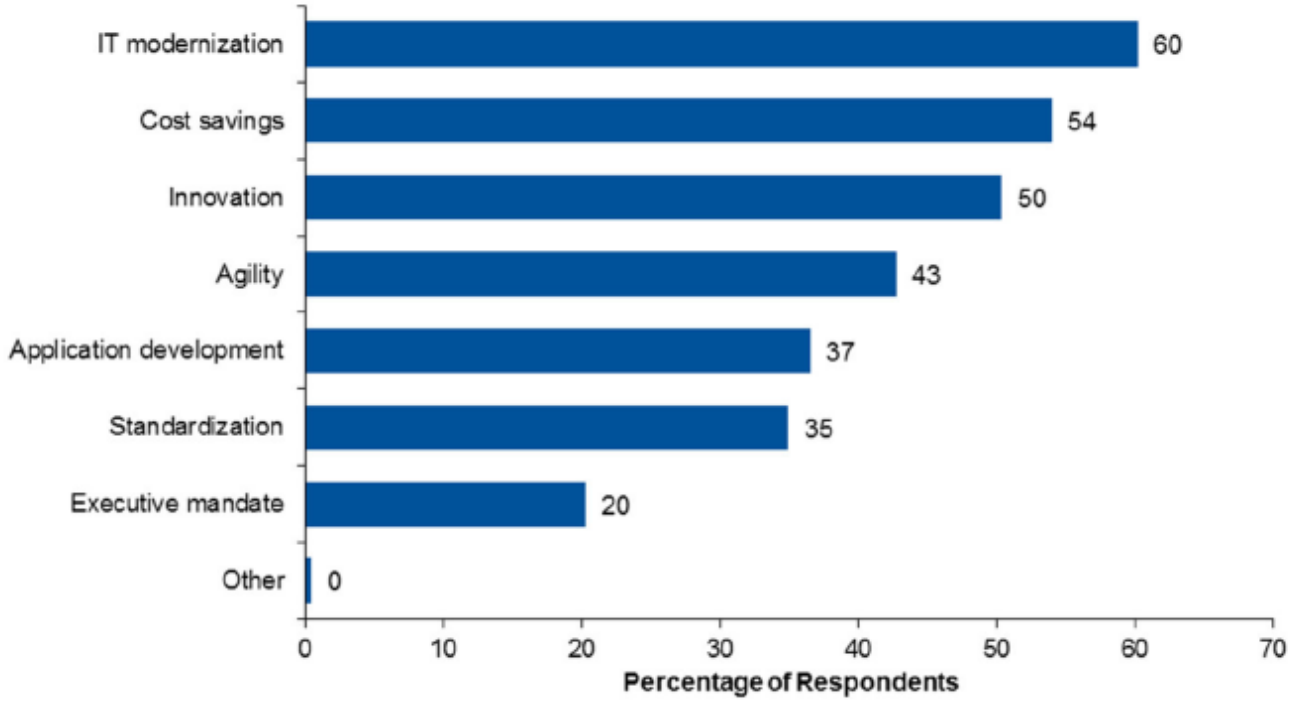

Figure 5: Top reasons for using or considering the use of Public Cloud Services – Gartner 2016

2.2.1 Reduced Cost

One of the reasons cited most often by consumers and those considering cloud is that it is more cost effective than the same capabilities delivered through traditional in-house IT deployments (Gartner, 2016). The ability to reduce IT costs is often one of the initial drivers for cloud adoption as seen in figure 5 above. Though there have been a number of studies warning against the blanket assumption that using the cloud is cheaper having considered all of the costs, migration, management, test and development etc. (Gartner, 2016), it is still one of the primary benefits stated and main reasons for cloud adoption. In a traditional environment, resources are typically bought and used solely by specific applications. The resources are scaled to meet the maximum estimated peak usage requirements so as to ensure there are no performance issues with the application. This

approach can create separated silos of IT which are typically underutilized and deployed with a large up-front cost.

Cloud computing can enable a business to acquire the IT resources they need when they need them and release them when no longer required. This reduction in the need for large capital outlay for IT projects means there is a lower cost to adopt changes and implement the required technology (Zhang, et al., 2010). The typically much larger scale of cloud deployments means there is a lower cost per GB or IOP which can typically be achieved than that of a standalone business / enterprise with a smaller more silo based approach. The preference of some businesses to consume IT in an Operating Expenditure (OPEX – expenses that come from the ongoing regular costs a company pays to run its business) vs Capital Expenditure (CAPEX – funds that businesses use to purchase major physical goods or services which they keep for a period of time) model. This OPEX model is also suited to cloud consumption and reduces the CAPEX to only the required infrastructure.

2.2.2 Faster Time to Value

In a traditional business the process of adapting or acquiring new IT resources is typically aligned to a number of procedures and restrictions. Traditionally, resource acquisition may take a significant amount of time which delays the development and increases the time to market. Companies are leveraging cloud to enable them to develop and release products and services in a much quicker and more consistent time frame. Cloud computing can provide businesses the ability to provision services automatically with technologies which monitor the requirements and deliver the required services such as auto-scaling (Amazon Web Services, 2017). This quick and granular reaction done automatically ensures that there is limited wasted resources and the delivered service is as efficient and effective as possible. The consumption of infrastructure, platforms or software as a service enables businesses to cut through the complexity of setup, configuration and management of the service, thereby lowering the risk and producing a certain outcome. This therefore reduces the process and time to provision the IT services and reduces the time to market overall (Marston, et al., 2011). This allows a business to respond in a more agile fashion to changing market conditions. This agility provides the business with the capability to develop and test on a constant basis and drive innovation and research and development into new products and services.

2.2.3 Scalability

There can be a significant cost implication for businesses if they are designing their infrastructure to satisfy the demands of peak workloads. This is more apparent when the overall peak duration is particularly short in nature and requires the business to provision and maintain this environment for these periods. This requirement for a larger environment to meet these short peaks therefore impacts and increases the overall cost of deployment for the business. Purchasing this hardware upfront and then leaving it idle for long periods

of time is a poor use of resources. The ideal situation is to be able to deploy and pay for the resources you need, when you need them, and release them when no longer required. This scalability is one of the key criterion in the definition of Cloud Computing provided by NIST (Mell & Grance, 2011). This ability for consumers to easily and automatically scale resources to meet workload demands means that Cloud Computing can be delivered in the precisely sized configuration required at the time. It therefore enables a more cost effective method to buy and release IT resources compared to the traditional method of upfront purchase and then its usage for a short or only specific period of time.

2.2.4 Flexibility

Another key aspect of Cloud Computing as defined by NIST is elasticity / flexibility (Mell & Grance, 2011). In cloud deployment models there is the flexibility for customers to request, provision, use and release resources and services as and when required. There is therefore a reduction in the requirement to run and maintain many aspects of the environment when fundamentally the purpose of cloud computing is to rent the services and resources required. An example of the flexibility available to cloud deployments include the ability to have more options for data protection and availability which can be adjusted through a management layer and not require physical changes or upgrades to the physical infrastructure as would be typical for in-house technology. Cloud instead provides choice for protection, performance and availability through the relevant consumption models of which cost is typically the determining factor relevant to the importance or value associated with the data. The further flexibility of cloud means that services can be acquired from within one cloud vendors' offerings or from multiple different vendors depending on their capabilities and options. Time and or money does not need to be spent on deploying additional capabilities when they can be consumed on a more flexible timeframe and cost from a Cloud Service Provider.

2.2.5 Innovation

Largescale surveys highlight that innovation or competitive advantage are cited as primary drivers to cloud adoption for 35% of respondents (NorthBridge / Gigacom, 2014). As can be seen in figure 5 above, innovation was cited by 50% of respondents to Gartner's survey of "Reasons to use or consider using Public Cloud Services" (Gartner, 2016). Innovation can enhance traditional IT environments in a number of ways. For example, developing and testing new applications in a production environment is not ideal as it can impact the live applications and brings a lot of risk. Typically development is done on a standalone pool of compute systems so as to reduce the risk to the production environment. Unless the business maintains multiple different configurations and OS types etc., the test configuration is typically it is not as robust as it could be. The ability to leverage a dedicated pool of compute systems in the cloud for short periods of time reduces management and overheads for the business. This allows more robust testing through the deployment of all relevant configurations, OS etc. In turn developers are able to spend more time working on and trialling different models or applications with the freedom and knowledge that they have not wasted valuable resources

on exploratory developmental areas. This freedom of configuration and reduction in the time and cost aligned to development ensures that time and resources are freed up for further innovation and application development.

A further benefit to the use of cloud as the base for the data sharing is innovation through collaboration. Due to the flexibility of access available to cloud platforms, collaboration between different business groups can easily be handled through the sharing and simultaneous access to the data. Users in different locations can quickly and easily work on and add value to a business' data thereby increasing meaningful collaboration and aiding innovation (Wang & Wang, 2012).

As well as enabling innovation in traditional environments, the availability of cloud computing additionally provides new formats and delivery modes for development of applications that were not available previously (Marston, et al., 2011). One example of the new technologies and capabilities delivered through cloud usage is the expanding platform of application and sensor communication being delivered across what is known as the Internet of Things (IoT). The gathering of this sensor data and the data analytics capabilities provided by Cloud platforms are an example of how businesses can use cloud to gain a competitive advantage.

There is an ever increasing amount of mobile applications interacting with each other and with the ever increasing data being gathered from the growing ecosystem of sensors which defines the Internet of Things (IoT) (Daniele Miorandi, 2012) (Andrea Zanella, 2014). All of the main technology analysts (Forrester, Bain, HIS, Gartner, etc.) are forecasting huge growth in the number of connected things in the IoT in the coming years (Forbes, 2016). For example, Gartner predicts 6.4B connected things will have been connected and be in use worldwide by the end of 2016, this is up 30% from 2015. They forecast that this figure will reach 20.8 billion by 2020. They also estimated that in 2016, 5.5 million new things will have been connected every day (Gartner, 2015). The accessibility and programmability of cloud based applications means that developing and leveraging the compute power available in the cloud to run data analytics is something that can be done very easily and produce many relevant outputs for different queries. Using predictive analytics, businesses are able to approach opportunities, risks, and customers differently because they have foresight they lacked previously. Doing this through the use of the cloud enables these new insights to be discovered in a shorter time (Talia, 2013).

One example of innovation put into practice is obvious when considering how Lufthansa is using an IoT-based strategy to create an entirely new business. This business consisted of mining data from their maintenance, repair and overhaul (MRO) operations and providing it to their customers. They are also using real-time aircraft, airport, and weather sensor data to improve on-time performance and optimize operations. Lufthansa aggregates all available data together to consistently deliver excellent customer experiences as a result (Shay, 2016).

2.2.6 High Availability & Business Continuity

Many medium sized customers struggle with the costs associated with implementing a robust highly available / business continuity strategy. Cloud computing has the ability to ensure resource availability at whatever level of service a customer's application requires or policies dictate. The cloud infrastructure consists of redundant hardware at every level and is protected with additional clustered and highly available software. This combination of technology enables large scale fault tolerance to be available within the deployment. The ability for cloud providers to protect data across the globe through the use of dispersed data centres and through the use of globally aware products at the backend ensure data is always available even in the event of a regional failure. The capabilities provided by cloud providers to enable the protecting of availability in this regard ensures that the company can reduce the necessity to spend additional money on their own data protection or business continuity infrastructure. This removes the requirement to set up a remote secondary site with the replication or business continuity technology required. The ability to consume availability / business continuity as a feature on a subscription means that consumers can protect the data which is most at risk or important, easily and cost effectively. This can help to mitigate the impact, from both a time and financial perspective, of downtime on a business.

2.2.7 Security

There are many known and unknown issues with security in the cloud. For many consumers they may envisage that the security / data protection capabilities available through the usage of the cloud would be better than they could realistically achieve / implement themselves without additional considerable investment in products and expertise. This can sometimes challenge the popular view of cloud as being a risky location for data security. However, many reports and technology analysts support cloud as being more secure for many reasons (Infor, 2015) (Gartner, 2014). There are many valid explanations why the cloud is more secure. Internal risk from malicious or unintentional loss or breach of security by employees being a primary concern. On cloud services, employees who may wish to cause intentional harm or breaches for a particular customer will find it more difficult to locate certain data in the cloud. Typically employees of the Cloud Service Providers also face tougher sanctions for any breaches. The Cloud Security Alliance (CSA) identify malicious insiders as one of the "Treacherous Twelve: Cloud Computing Top Threats in 2016" but there is no distinction or additional concern mentioned regarding externally located employees (Cloud Security Alliance, 2016). Cloud businesses have to build secure data centres that are independently audited, adhere to standards such as Soc 2 Type II (a company with this certification has proven that its system is designed to keep its clients' sensitive data secure) (SSAE, 2016), and are used by hundreds to thousands of tenants. This is a requirement from a legal perspective but also a business perspective. The CSP's business existence depends on their ability to provide a secure environment to their customers. One good summary is provided by Frank Gillett, Forrester Research "*Using cloud services means that your data is better protected than if most of us tried to manage it on our own…. Large-scale services are all much better than we are at avoiding data loss from gear failure,*

keeping software up to date, upgrading hardware, and constantly improving security" (Wall Street Journal, 2014).

The knowledge that enterprises IT departments may achieve better security through the usage of cloud is combined with the realisation that the data security would be primarily outside of their control. This loss of total control over the data is something which could have implications in the future, most often around the ability to meet compliance and changing privacy requirements. For larger enterprises who would have the expertise on hand to protect their data to a realistic standard would be more concerned with the looser control and standards that they would be relinquishing. Enterprises are considering a more hybrid approach and the movement of additional data workloads to the cloud as the security of off-premise clouds is being invested in and strengthened significantly (Everest Group, 2014).

The importance of the role of security with regards to maintaining the privacy of data is well summarised in the following quote: *"You can have security and not have privacy, but you cannot have privacy without security"* (Mather, et al., 2009). In order to protect the privacy of the data the key concern should be to ensure that all the regulatory requirements are being met from a security point of view. Once the security and privacy requirements can be met, then the deployment model is not important. If a cloud deployment can provide the same or better capabilities than in house configurations, then it should be considered. When this security is guaranteed and the business is satisfied with its protection strategy then the location of the data is no longer a concern. This allows the positive aspects of security provided by cloud computing to be embraced.

2.3 Adoption of Cloud Computing in Industry.

There has been a significant uptake in cloud computing for general consumer workloads, test and development and software as a service delivery of email / CRM applications. Public and private / hybrid cloud adoption has increased significantly in the years between 2011 and 2015, +43.3% and +19.2% respectively (North Bridge, 2015). Countering this adoption, as mentioned previously, there has been a significant hesitation in industry to leverage off-premise cloud technologies (Public or Private) (Sultan, 2011) (Yeboah-Boateng & Essandoh, 2014). This is especially accurate for workloads which are deemed of greater value / importance. This section will present a review of the adoption of cloud relative to the different cloud computing platforms of IaaS, PaaS and SaaS. The purpose of this review is to highlight where enterprises are adopting cloud and where the research review should focus.

2.3.1 IaaS

Infrastructure as a Service (IaaS) continues to grow as enterprises move away from data centre build outs and refreshes and more infrastructure moves to and is deployed in the cloud. It has increased from 11% of

enterprises leveraging IaaS in 2011 to 56% – in 2014 (NorthBridge / Gigacom, 2014) and continues to grow as the latest iteration of the survey showed that IaaS is in 67% of surveyed businesses in 2015 (North Bridge, 2015).

IaaS is the cloud delivery method where there is significant differences in the consumption options available. Public IaaS (infrastructure delivered as a subset of shared resources hosted by an offsite third party), Private on premise IaaS (infrastructure delivered on exclusive hardware within an onsite datacentre) and Private off-premise IaaS (infrastructure which is not shared but delivered offsite in a hosted location) can all be delivered and consumed in different fashions. PaaS and SaaS are typically delivered by a third party vendor and as such are hosted offsite and would therefore all be typically be classified as Public Cloud offerings. When IaaS is referred to and thought of it is typically understood to be Public Cloud IaaS as it makes up the most consumed offering. That which is located and hosted by a provider for multiple tenants to consume. There are various methods that vendors use to logically separate the data but it is very much a Public Cloud IaaS. Amazon VPC (Virtual Private Cloud) is an example of this, it is a logical isolation of the AWS Cloud (Amazon Web Services, 2016). Cloud consumers have different platforms and applications/solutions available to them and many of those being deployed on IaaS would not be considered typical enterprise applications. As mentioned, IaaS consists of definitive separations of consumption. Private and Public.

The expectation from research firm Gartner, is that the highest growth in the public cloud services market will come from IaaS. The projection is that it will grow by 42.8% in 2016 (Gartner, 2016). Reinforcing these findings, another research firm, International Data Corporation (IDC), found that almost two thirds of IT organisations are either using or planning on using public cloud IaaS by the end of 2016 (IDC, 2016). Additional forecast from IDC is that as the public cloud IaaS market grew 51% in 2015, the high growth will continue through 2016 and 2017 with a CAGR (compound annual growth rate) of more than 41%. This heightened adoption and growth will further increase the migration of workloads from traditional environments to IaaS.

2.3.2 PaaS

PaaS has grown its customer base from 7% to 41% of enterprises between 2011 and 2014 (NorthBridge / Gigacom, 2014). This growth continues to be driven by the willingness of customers to forego control and responsibility for deploying and maintaining hardware and the platform and instead concentrating on the development and deployment of relevant revenue generating business applications. The adoption of PaaS has been somewhat slower than that of the combined IaaS offerings and SaaS in general. Industry commentators and reviewers attribute this to a number of factors. Cited is the confusion in the definition of PaaS and the blurring of lines between IaaS and PaaS provided by cloud providers such as AWS which means there are a huge number of varieties and “flavours” of PaaS (Kavis, 2014) (Natis, et al., 2011). This creates confusion for consumers and also means that surveys regarding the usage of PaaS may not be responded to accurately. As

PaaS is a newer offering and has less deployment and operational features available, the adoption of PaaS has been highest in smaller more agile companies where development can be done more quickly and without the requirement to manage a datacentre. Large enterprises on the other hand have a requirement to have largescale testing completed, a guarantee that SLAs will be met and as their workloads are currently in-house in a data centre would prefer a more dedicated environment in which they know that their languages are all going to work effectively. The expectation from recent industry surveys is that PaaS adoption will continue to grow. It is also expected that when reviewed in future surveys the adoption by enterprises will show as significantly stronger from 2015 onwards as the services strengthen and some clear leaders in the space emerge. This is backed by the fact that PaaS revenue CAGR is the highest of the deployment models with 38%, albeit from a lower starting point (North Bridge, 2015).

2.3.3 SaaS

Adoption of SaaS by industry has increased most significantly from 13% in 2011 to 72% in 2014 (NorthBridge / Gigacom, 2014). SaaS adoption continues to grow but as it is currently generating the largest revenue, its CAGR is understandably the lowest at 18% (North Bridge, 2015). SaaS adoption continues to be driven by the increase in availability and ease of consumption of public cloud based software offerings of typical on premise licensed solutions. As the continued proliferation of mobile technology means that workers and enterprises want and need to access more data from remote locations, the desire to access applications via SaaS continues to be more compelling. Enterprises have begun consuming a huge amount of SaaS offerings over the last 10 years. This adoption has been driven in large part by Salesforce.com's successful IPO in 2004 which validated the model in a way previously not seen. The benefits of SaaS consist of the largest overlap to the benefits of cloud adoption in general. The reduction in cost and the increase in flexibility and scalability of the service along with quality guarantees are all factors that significantly influence an enterprises decision to adopt SaaS (Benlian & Hess, 2011).

2.3.4 Adoption of Platforms Summary

Each of the platforms has its own strengths and weaknesses which encourage or restrict the current and future adoption rates in enterprises. SaaS has the highest reach due to the ease of consumption. PaaS has the highest CAGR based on an adoption and understanding of its usage and benefits across industry. The adoption of IaaS is the most significant to this research however due to its large current consumption and predicted continued adoption going forward. Research by Gartner predicts that between 2015 and 2020 IaaS will be a close second to SaaS for the amount of revenue spent on it but will have the highest CAGR over those 5 years (Gartner, 2016).

Further research performed by Statista predicted that in 2016, spending on public cloud Infrastructure as a Service hardware and software is forecast to have reached $38B, and further growing to $173B in 2026. SaaS

and PaaS portion of cloud hardware and infrastructure software spending are projected to have reached $12B in 2016, growing to $55B in 2026 (Statista, 2016). The aggregation of these reports highlight the continuing pace of IaaS adoption and its importance to the overall public cloud landscape.

Figure 6 below provides an overview of predicted spending on public cloud infrastructure worldwide from 2015 to 2026 based on Statista modelling (Forbes, 2016).

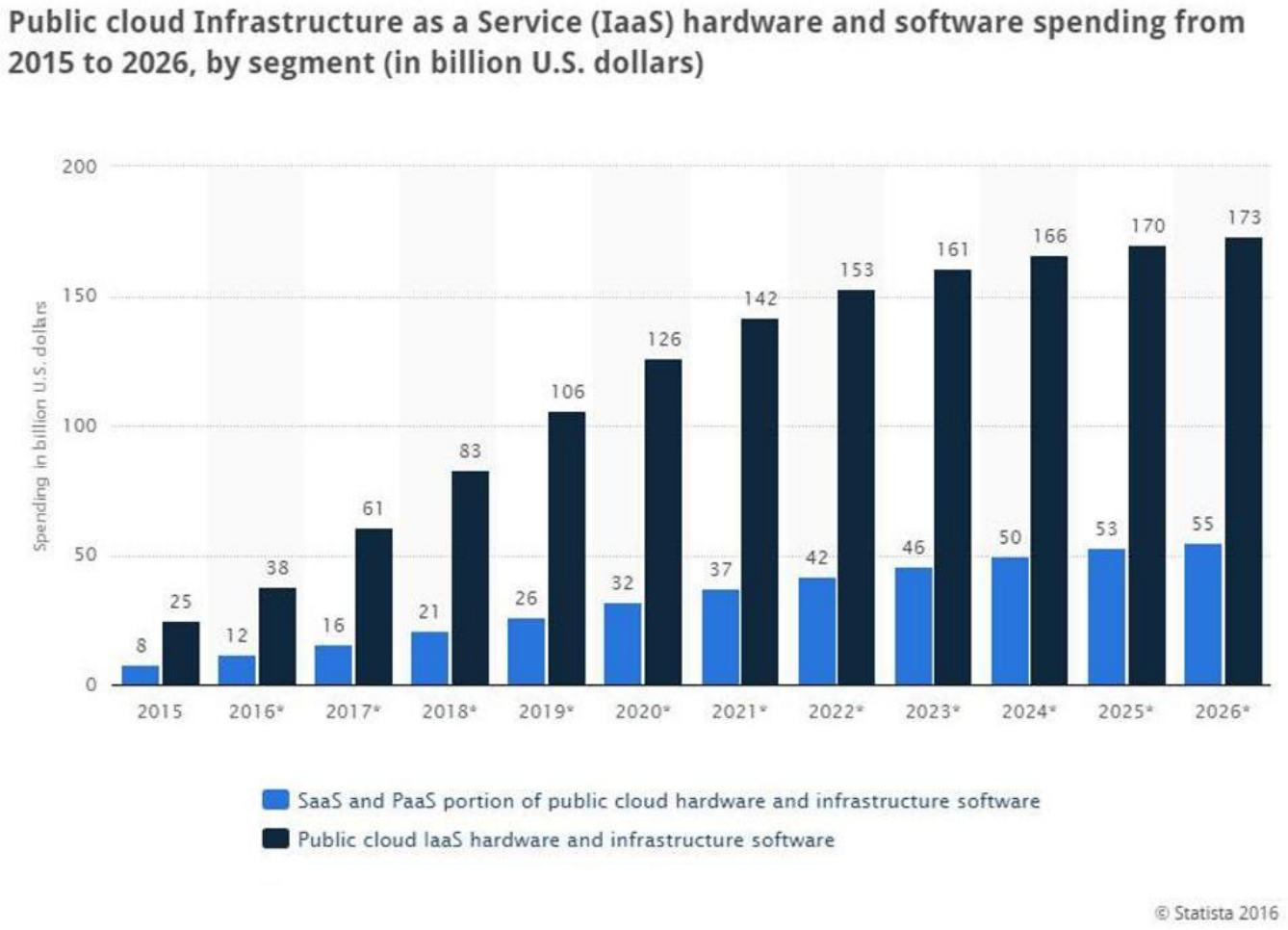

Figure 6: Public Cloud IaaS spending from 2015 to 2026 – (Statista, 2016)

2.4 Factors Influencing Cloud Adoption

Cloud computing is gaining significant investment as the result of widespread adoption success stories. However, fear and general concerns about some factors such as relative cost and trust — combined with the confusion caused by "cloudwashing" (using the "cloud" term to confuse the definition and make claims of cloud that are unsupported) — sometimes counteract the enthusiasm and excitement. The main factors influencing the potential adoption of off-premise cloud technologies for IT workloads have been comprehensively surveyed and reviewed in a number of papers. These inhibitors are similar across all cloud

platforms, IaaS, PaaS and SaaS. One set of findings discussed the decision to select a cloud service provider based on six main requirements; flexibility, costs, IT security and compliance, scope and performance, reliability and trustworthiness, and service and cloud management (Repschlaeger, et al., 2013). Another study found that security, performance, availability and portability were the key concerns (Everest Group, 2014). A KPMG study into global Governments highlighted similar concerns and challenges with adoption factors such as response time or the ability of cloud to provide required level of performance, security, difficulty integrating or being able to realise the benefit of cloud, data security and privacy are seen across governments (KPMG, 2012). UK IT and business professionals' survey similarly outlines the perception of suitability for cloud adoption (Virtustream, 2014). The main factors which have been seen to influence the adoption of cloud are summarised in the following sections.

2.4.1 Availability & Performance

The access to data and services is fundamental for business transactions and processes. Availability in this scope is concerned with uptime of the service and reliable performance. Without an agreement on both uptime and reliability there can be no confidence that the requested service will be provided. This in turn creates hesitation among potential adopters (Stratogen, 2015). Typically any disruption to a known Cloud service will affect a greater number of customers and so will be reported many times over in the media, even though the downtimes and impact to customers are typically very small in comparison to a typical enterprise IT infrastructure. This issue of availability perceived by decision makers can be compounded by the risk of outages occurring for non-technical reasons such as the provider being suspended or going out of business (Armbrust, et al., 2010).

There has been some hesitation around the adoption of IaaS for enterprise / high performance computing requirements. The performance of public cloud technologies like Amazon EC2 has been found to perform poorly in comparison to modern HPC (High Performance Computing) systems (Jackson, et al., 2010). Known issues and "resource-freeing attacks" on shared cloud infrastructure place additional uncertainties on the performance which is available and can be guaranteed to a cloud consumer (Varadarajan, et al., 2012). Isolation techniques have been investigated to ensure fairness of usage (Krebs, et al., 2014). There has been significant progress made with all aspects of cloud technologies in the last number of years. Items such as AWS Virtual Private Cloud model are developed with the premise of resolving many of these performance issues. A recent study by Gartner would suggest that the performance is still a key consideration for cloud consumers however. The study highlights that consumers select the following as a reason for not migrating to public cloud: "Most of the x86-based applications do not perform well under virtualization, or have architectures that are not well-suited to a cloud IaaS environment." (Gartner, 2016).

2.4.2 Portability and Interoperability

This is a concern which addresses the primary two areas of three. Cloud portability, interoperability and compatibility are closely related but should not be confused as they relate to different things. Interoperability is the ability for cloud to work together, wherever the clouds are located there should be a means for them to interact. Compatibility and portability refer to how they interact. Compatibility meaning that they work in the same way regardless of location and portability is the ability to easily move and reuse the instances and data no matter which cloud provider is being considered (Zhang, et al., 2013). The first being the potential difficulties integrating the cloud with current enterprise systems and the second being the ability to move data once it is resident in the cloud. Many documents and surveys highlight integration and interoperability and portability as key challenges that customers are facing (KPMG, 2013). Similarly ENISA (The European Network and Information Security Agency) and the European Commission have recognised that Cloud vendor lock-in is a high risk aspect of cloud infrastructures (ENISA, 2009).

Some organisations need to integrate cloud service with in-house systems and create a hybrid solution. The successful interoperability between private and public clouds are key considerations to ensure there is widespread adoption of cloud computing by the enterprise (Avram, 2014). Initially the enterprise must ensure that their environment is modernised and standardised to ensure the processes, data and systems are all streamlined. The use of scalable infrastructures and integrated software allows a consolidated platform and management instance to be created for many technologies which helps with the integration between onsite and offsite. The level of integration possible depends on the applications and the manner in which they can be adapted to changing environments. For enterprise applications this level of interoperability is either an enabler or an inhibiter to integration with cloud platforms. The integration of data storage and transactions for these services is also typically time consuming and costly and the implementation of this integration can also be a barrier.

Regarding portability, the use of SaaS, IaaS or PaaS models can become a potential issue as the customer becomes more and more dependent on the proprietary service. *"Cloud computing has become a lot like the Hotel California: Once you pick a provider you can check out anytime you want – but you can never leave"* (Haislip, 2012). The cloud computing community does not have a defined set of interfaces or a universal set of standards which can result in an increased risk of vendor lock-in and so provides a barrier to adoption (Lewis, 2013). The ability to quickly and easily move and reuse a company's data assets from one service to another is not always possible or clearly documented due to proprietary APIs, or if it is possible, the costs associated may be prohibitive (Phaphoom, et al., 2015). This removes the ability for potential consumers to take advantage of the availability of different features of cloud vendors as the services change and means that consumers are wary of potential lock-in to a service. Portability is a concern more so for customers who are unsure of their growth rates or performance requirements. This uncertainty means that they may not be able to clearly envisage what the costs and impact to their environment would be if their environment or the cost of

running it in the cloud change more than expected. An unpredicted spike in requirements or costs for the service would impact a business's ability to execute effectively. If there was a lack of portability, or a an un-budgeted cost for migrating the data to another service provider or back in-house then this lack of portability will cause an issue for the business.

2.4.3 Migration Complexity

Even with the advent of additional service and applications to assist with migrations there is still a large amount of risk and complexity involved. There has been a certain amount of research carried out into Cloud Migration technologies recently and a systematic review (Jamshidi, et al., 2013) highlighted a number of areas for improvement and gaps with regards to migration tools and frameworks. Frameworks available include Cloudstep (Beserra, et al., 2012) and extensions to the Darwin framework (Ward, et al., 2010) which highlight the impact of manual vs automated migration and the risks associated with migration to the cloud. The cloud hype that cloud consumption is simple has been replaced with the reality that each and every application must be examined and that meaningful external help is needed to ensure that the business is successful in cloud migration (Accenture, 2016). Planning for migration should not be underestimated as the impact to the enterprise needs to be assessed and considered prior to cloud adoption. Lack of expertise has been cited as a reason preventing adoption of public cloud in a number of surveys and papers. It was found to be the 3rd highest factor in the most recent North Bridge survey of IT decision makers (North Bridge, 2015). Due to this complexity of migration, there is an additional cost associated, including both time and expertise which negatively affects the likelihood of cloud adoption (Nuseibeh, 2011). Gartner highlighted the fact that many organizations require and benefit from the use of a Managed Service Provider (MSP) to provide operations management for a number of years post migration to ensure the best value is achieved (Gartner, 2016). There is a focus on the delivery and development of cloud migration tools to aid consumers in the migration of their workloads to and from cloud deployments to ensure that the process is simplified and removed of the potential for errors when performing manual scripting and migrations. There is still an issue that the tools are not available or manageable for more complex workloads and so a large amount of consultations may be required to enable successful migration (Gartner, 2015).

2.4.4 Security

Security is an important consideration irrespective of where the data is located. Surveys have consistently highlighted security as one of the top three concerns with cloud service models (Repschlaeger, et al., 2012). There is an advantage gained from the use of cloud technologies as there is additional protection from the scale and the ability of the provider to implement additional security measures not possible in the traditional environment of a typical cloud consumer. Depending on the cloud model being considered there are different levels of security which can be managed by the business and which are the responsibility of the service provider. There is a difference between security of the cloud, being the security measures the cloud service

provider implements, and security in the cloud which is relative to the security measures that the customer implements and operates regarding the customer content and applications that use the underlying cloud services. For comparison, the use of an IaaS platform by a business means that they are responsible for everything, applications, data and VMs through the services controls. Everything apart from the security of the underlying hardware. The service provider is responsible for the security of the data centres, network and systems. On the other side, a SaaS consumer is only responsible for the access to the software. The underlying hardware, platform and applications security is the responsibility of the service provider. Most cloud service providers have documentation outlining where the responsibility for security lies and what areas their customer should secure themselves (Amazon Web Services, 2016).

Perceived risks are magnified as having data hosted on a cloud service provider means that a business whose data and environment may not previously have been an economical target may now be more prone to attack. The Cloud Security Alliance identified the fact that Cloud providers are highly accessible and the vast amount of data they host makes them an attractive target as well as the fact that the shared resources means shared vulnerabilities. These security concerns are apparent in a number of their "Treacherous 12 Cloud Computing Threats", primarily Data Breaches (Cloud Security Alliance, 2016). The Cloud Service Provider (CSP) provides a larger infrastructure which can act as a much more significant target for hackers. The fact that there is a larger footprint with more data it may be more economical for attack due to the ability to access many different consumers data in one go. The other threats such as malicious workers, other consumers who may access data across barriers, etc., are increased. Some of these threats are increased as the control over decisions, such as worker selection, is with the Cloud Service Provider (CSP). Many of the risks and benefits are outlined in the ENISA document on the matter (ENISA, 2016). There are a lot of known issues and vulnerabilities with cloud technologies and securing them requires much consideration and planning. Data security, administration and control, logical access, network security, physical security, virtualization and compliance are all areas of security which need to be accounted for, as is apparent from the many recorded security breaches. Similar to data privacy, the relinquishing of total control over the companies data is an inhibitor to adoption of cloud services. To mitigate risks to security of the data in the cloud, appropriate processes need to be developed and followed to ensure that the risks and controls are identified and that best practices are adhered to. This helps to ensure necessary protection and actions are identified to ensure the business and IT objectives are met (Carroll, et al., 2011).

2.4.5 Data privacy and legal concerns

Depending on the organisations operating function, healthcare, government, finance, etc. there are different amounts of responsibility attached to the data. The organisations may have multiple different regulatory and compliance considerations when dealing with their users data. Some of the main data protection requirements are those outlined in HIPAA (U.S. Department of Health & Human Services, 1996), PCI-DSS (PCI Security Standards, 2016), EU Data Protection Regulations (EU, 2016), which all need to be considered when handling

customer data. Data privacy has been ascending the list of concerns of consumers throughout the last number of years as more and more breaches of security have enabled a breach of privacy. There have been over 5000 data breaches made public since 2005. Some of the most well-known breaches such as the following have all lead to this increase in concern: 2007: Google Web Service, Salesforce.com; 2009: Microsoft/T-Mobile; 2010: Hotmail; 2014: Sony hack, iCloud celebrity photo leak, Heartbleed; 2015: Microsoft / Xbox One, Uber; 2016: Yahoo, World Anti-Doping Agency, Hillary Clinton campaign. (Privacy Rights Clearinghouse, 2016)

Similar to the control over security of the data, depending on the cloud model adopted there is also different responsibilities when it comes to the responsibility for the privacy of the data. IaaS platform means that they are responsible for everything apart from the protection of the underlying hardware. This includes being unaware of the application level and being able to ensure data-level compliance such as geographic restriction of data etc. Alternatively a SaaS consumer is only responsible for the access level of the software and control over the data which is given to the SaaS but otherwise needs to have confidence in the provider's responsibility and compliance. They need to be sure that data is located in certain geographies and not moved without their knowledge, who will have access to the data, what will happen in the event of a data breach, how will the data be destroyed, etc. The cloud consumer has some methods to vet and verify the capabilities of the service provider including advice and baselines for the arrangement of Service Level Agreements (SLA) and Privacy Level Agreements (PLA) provided by the Cloud Security Alliance (CSA).

When the data is kept in-house there is total control over the location, storing, and processing of the data. As companies are inevitably responsible for ensuring the security and integrity of the data they have gathered and are using, the outsourcing of services to the cloud doesn't change their responsibility or legal obligations. The landscape of data protection is changing, new stricter requirements and potential large fines and prosecutions for non-compliance is becoming more apparent with reformation of regulation such as the EU Data Protection Rules (EU, 2016). All of this change currently taking place and additional legislative amendment on the horizon results in uncertainty for cloud adopters. Uncertainty as to whether the cloud computing model provides sufficient safeguards to protect the privacy of their customers' data. Without confidence they cannot ensure they are not in violation and at risk of prosecution and as such the management of this data privacy is an inhibitor to cloud adoption. This loss of control and reliance on enforcement of these regulations by a third party can therefore be seen as another reason why data may remain in-house and why it is becoming more and more of a key challenge that customers are facing with adoption (KPMG, 2013).

2.4.6 Cost of Adoption

Customers are aware of the potential benefits in adopting cloud computing. Cost reduction is key and was cited as one of the main reasons for cloud adoption. However as can been seen in a recent Gartner report (Gartner, 2016), unless the existing environment is very inefficient or a migration is going to be required anyway, an unscheduled migration to cloud platforms is unlikely to result in much cost savings. One of the

key business drivers for adopting cloud business cost reduction, it is also one of those which is least often accomplished (Accenture, 2015). Since cloud has become more embedded in businesses and adopted in varying degrees there has been a change in the considerations regarding the cost of the cloud and the realisation that it is not just a cost per GB or per IO. Measuring ROI (Return on Investment), high cost of implementation, and lack of clarity of the total cost of ownership all appeared as some of the most challenging areas when adopting cloud in KPMGs Cloud Survey Report (KPMG, 2014). Another KPMG study highlighted that the survey respondents identified cost as being one of the key challenges with their approach to cloud adoption. Implementation/ transition/ integration costs being too high were the most cited (KPMG, 2013). This appears to be aligned with the ongoing attempt to drive integration with consumers existing environments and the realisation that moving data to the cloud is not as easy as expected. Both of these appear to change the time, effort and requirement for external assistance which then impacts the overall cost for migration to and adoption of the cloud service. Gartner recognise this and provide analysis based on the fact that 80% of organisations doing a "copy and paste" of internal business to public cloud IaaS will not achieve meaningful cost savings. There is some guidance and analysis provided on how best to calculate if it is possible to save money by migrating to cloud IaaS and advise that there are a number of key considerations such as scenario based modelling, modelling of financial implications over time, and, evaluate the benefits and costs across the business (Gartner, 2016). In all instances additional connectivity bandwidth may need to be leased / consumed to meet performance needs. There is technology available to optimise the WAN connectivity to increase the throughput and decrease the latency with varying levels of requirements being catered for. The leaders in this space being Cisco, SilverPeak and Riverbed (Gartner, 2016) but there is the cost implications of deploying technology like this also which can add to the overall cost analysis. The cost of data integration and migration to the cloud can be substantial as there are different requirements and APIs for different clouds and as such there is no straight forward method to compare the overall cost for adoption. Add to this the potential requirement for the consumer to remove or move their data from the cloud service and the costs and charging model which may be associated to this and how it is implemented by different providers it is easy to see how this has risen through the ranks of potential barriers to adoption.

2.4.7 Factors Influencing Cloud Adoption Summary

Each of the topics previously discussed have an impact on the likelihood that businesses will adopt cloud for their particular use case. Considerations will be made based on both the probability and weight applied to each aspect. Decisions based on many aspects will need to be deliberated. Many factors need to be considered such as whether the data will need to be migrated, removed, protected from privacy or security breaches, interact with other aspects of the business or cloud platforms, its availability requirements as well as the performance and costs which are desired and deemed acceptable.

Many of these factors will be dependent on the underlying businesses value of the particular data and as such are harder to quantify and measure. Security for example is very important for businesses data but can be very

hard to evaluate. How can a company be sure of the security of their data and the cloud based applications on third-party infrastructure? How can this be easily measured and compared? There are some attempts at providing benchmarks to enable easier comparisons available. Businesses need to find means to investigate what tools are available to assess privacy of user data in the cloud (Gao, et al., 2011).

As per the earlier overviews, a number of areas are simultaneously highlighted as the primary benefits to cloud adoption while also being deemed potential inhibitors. Two of the most apparent, quantifiable and frequently touted as core reasons for cloud adoption are those of performance and cost of adoption. This ambiguity raises a number of questions around the means and ease in which businesses can weigh up the adoption of cloud. Decisions need to be made based on accurate details around both the advantages and disadvantages.

Taking performance as a key principle of cloud computing adoption. One of the benefits of cloud computing is its scalability, this is apparent and similarly applicable for both small (<1000 employees) and large (>1000 employees) companies (Accenture, 2015). The question is then, why, when one of the key benefits of cloud computing is its scalability and access to unlimited resources, is performance also referenced as a key inhibitor to adoption. As the cloud scales, so too should the performance.

Similarly for the cost of adoption, as the key benefit of cloud is scalability and the ability to consume resources on demand, why is there such a hesitation around cloud adoption due to cost? When just the appropriate amount of resources for the environment desired can be consumed on demand, there should be a significant difference in the cost of running particular workloads in the cloud. This would seem to suggest that there is an issue with accurately quantifying the costs and comparing the efficiencies of running workloads on cloud resources versus the status quo. Often cost can be a negative effect experienced by business. This can be due to their unconsidered adoption of cloud for its openly touted potential benefits with a lack of sufficient impact analysis. Risks not considered have the potential to undermine any and all of the benefits which were anticipated (Walterbusch, et al., 2013).

The continuing desire to reduce cost and increase efficiency is the primary organizational improvement that companies want to realise through cloud adoption (Accenture, 2015). This paper proposes that the combination of performance and cost efficiency and the benefits or disadvantages of running workloads in the Public Cloud should be accurately reviewed.

2.5 Review of Cost Analysis and Performance of Cloud Computing

This section will present a review of the general research work which has been completed on both the performance and cost of cloud based systems.

The cost of adoption of new technologies encompasses a large number of items. Many of these are not immediately obvious when businesses are doing their cost / benefit calculations and so can sometimes be ignored. Some costs which are not always apparent are items such as support, time spent by developers on application and data modifications, insurance against data loss, as well as additional disaster recovery costs (Avram, 2014). It can be extremely difficult to determine the potential impact the migration of workloads to cloud platforms has on the aforementioned hidden costs. These costs, as well as the cost for personnel, training, services and time are hard to determine and are often unique to each business.

Some companies will focus on return on investment (ROI) calculations in cloud computing. Research has already been undertaken in this area (Misra & Mondal, 2011). The purpose of the research being to derive the return a firm would get based on their required investment in the cloud. However typically the calculations are lacking a complete breakdown of all of the components used in the ROI calculations. A more detailed analysis of each of these ROI components provides a more accurate comparison of the costs involved in remaining in-house or moving to the cloud. Having a detailed review and breakdown of the infrastructure costs, data patterns and project specific costs have all been found to significantly influence the cost-benefit analysis results. Having the required level of information detail can ensure that an accurate and informed decision is made. There are different models which provide guidelines as to the potential savings that can be made when moving from legacy systems to cloud deployment. They provide guidance on what to expect relative to the size of the enterprise (Nanath & Pillai, 2013). Many cost / transaction type comparisons have been made and there has also been an attempt to provide more relevant cloud performance, capability and productivity metrics to enable more accurate cloud evaluations. Cost comparisons of the impact of auto-scaling, mixed instances, quality of service (QoS) and different workload benchmarks on the potential economics of cloud adoption has also been performed (Hwang, et al., 2016) (Hwang, et al., 2014).

The performance of cloud based systems and attempts to accurately benchmark the numerous options is similarly difficult to precisely determine. Benchmarks are tools which enable the following question to be answered, "*What is the best configuration in a given domain"* (Folkerts, et al., 2012). Benchmarking is done with the intention of generating a report on how well different systems operate while gaining insights into any potential bottlenecks. For example, the SPEC CPU benchmark enables investigations into what the best CPU is (SPEC, 2017). The TPC benchmarking tools provide a means of evaluating the performance of different systems under the relevant workload. The TPC benchmarks provide various workloads that define different real-world application scenarios (e.g. TPC-C for OLTP (TPC, 2017) and TPC-DS for decision support systems- including Big Data (TPC, 2017)). Based on the application that the customer wants to test and therefore the workload being emulated, the goal is to reproduce some realistic scenarios and gather the results for comparison. Based on the metrics gathered from the benchmarking runs, the customer knows what to expect from the test system. The system owner can then make an informed decision to either change the test environment to investigate if any improvements can be made or if another configuration is necessary.

Benchmarking has been done across many different workloads and cloud deployments with various different tools and approaches (Cooper, et al., 2010) (Folkerts, et al., 2012) (Chhetri, et al., 2013) (Binnig, et al., 2009). As per the title of the following paper, *"A theoretical evaluation of what cloud benchmarking should, can and cannot be"*, an investigation and analysis of different methods is performed in (Folkerts, et al., 2012). This paper attempts to put some guidance and structure into benchmarking tools to reduce the variance and determine the most appropriate metrics for cloud use cases. As there are so many different considerations and variables to be selected from cloud technologies it often causes potential cloud adopters to struggle with determining what are the realistic performance expectations. This difficulty preceded the creation of many different cloud benchmarking platforms. The purpose of these cloud benchmarking platforms is to enable decision makers to make informed decisions about workload migrations (Chhetri, et al., 2013). This includes a review of different benchmarking tools and the workloads they represent with the purpose of identifying where any significant value can be added and outline the challenges of building tools to measure and compare them. Smart CloudBench is one example of such a platform whose design allows the automated execution of benchmarks on different IaaS clouds. It enables teams to compare representative load conditions to quickly estimate their cost/performance levels and to do a thorough evaluation of multiple cloud platforms. The conclusions found that higher price paid does not necessarily equate to better performance and the use of an automated benchmarking tool could enable enterprises to make more accurate decisions (Chhetri, et al., 2013).

What is apparent from the research papers reviewing cloud benchmarking technologies previously mentioned, is that there is no definitive recommendation as to the best approach or tool to use for any particular workload. The tools available and the workload being considered for cloud deployment are the primary concerns. It is necessary to have these decided before being able to determine the correct application setup to benchmark initially.

One such consideration which is under renewed analysis, is whether scale-out is truly better than a well-considered scale-up implementation. One such example is equating both physical and cloud implementation of both the scale-up and scale-out configurations for Hadoop workloads. There is a case for a review of the typical approach that a scale-out strategy is the best option from a cloud deployment perspective (Appuswamy, et al., 2013). The purpose of both the physical onsite and cloud based implementations was to challenge the general perception that scale-out is more appropriate for a Hadoop workload typically. The conclusion that scale-up was both competitive and significantly better than scale-out across all tests carried out defies regular perceptions. Further research with additional comparisons performed between larger numbers of lower powered, and smaller numbers of higher grade nodes have been made. These scale-up and scale-out type deployments are tested to investigate the different capabilities and benefits of each. The different contexts applicable to the systems under test, web-search (Vijay Janapa Reddi, 2010), and key-value pairs (David G. Andersen, 2009) provide opposing views on which configuration is better. The resonating conclusion is that there are differing abilities and benefits for each deployment but overall they highlight that the correct choice is workload specific.

Building on the information reviewed, the thesis proposes to contribute to the current state of the art. This will be done by performing comparative testing between scale-up and scale-out in cloud based systems and progress knowledge in this area. The comparisons of each will add to the work previously carried out by replicating as closely as possible the test implementation. There will be some further knowledge developed through the addition of supplementary considerations. The first consideration which will be added will be the comparison of scale-up and scale-out configurations against the same application. The use of the same application for both configurations reduces the requirement to use experiential knowledge when comparing different results. Using the same application means that the same benchmarking tool can also be used and the same results gathered. Previous research works have normalized metrics to compare different workloads performance gathered with different benchmark tools (Hwang, et al., 2016). Providing standardised results enables a more straightforward comparison using the metrics for both deployment models. When the results have been derived from the same benchmark tool they provide more directly comparable findings. Using previously researched cloud performance and productivity metrics such as cloud efficiency and productivity (Hwang, et al., 2016), this paper will provide comparisons between multiple deployed configurations. The metrics gathered will provide an accurate means to compare relative performance and cost of the different deployments.

Investigations into the impact of leveraging multi-threaded benchmark applications will also be carried out. It is not apparent that this has been considered in the comparisons of previous benchmarking runs. Why benchmarking with multithreads may impact the results in a positive or negative manner will be reviewed. The expected performance implications depending on the configuration of the cloud based system and the increased number of threads will be measured and reviewed.

This will provide a foundation for future analysis to be performed on an expanded set of application workloads and cloud platforms. The intention is to provide detail for cloud adopters and decision makers to consider their concerns around the influence of the metrics as inhibitors to adoption and enable a more educated choice.

3 Empirical Research Methodology

3.1 Research metrics

As discussed previously, the intention is to provide a comparative analysis between scale-up and scale-out configurations of cloud instances. The different configurations will be benchmarked to gather a range of relevant metrics, the purpose of which will be to enable an accurate comparison between each of the strategies. This will ensure that, as per the scope of this research, the performance and workload costs of both configurations can be reviewed and reasonable conclusions drawn. The interest in this comparison is due to the limitations which occur with instance sizes, whether they are physical or cloud based. Different workloads / jobs have different limiting factors. For example, CPU bound jobs are limited by the performance capabilities of the instance or the performance sum of the instances. Memory-bound problems are limited by the total memory (including cache) allocated within the machine instances. Whereas network latency and disk storage / I/O bandwidth limit storage-bound problems (Hwang, et al., 2014). In this paper the latency and throughput will be measured and are key to the cloud metrics described in this chapter. Observations around the CPU and memory usage on the running instances will be made but will not be key to the overall paper.

The ability to increase the size or number of instances therefore provides the ability to increase the overall performance potential to meet different workload requirements effectively. The cloud provides resources for consumption based on different instance types and required quantity of instance. For this research paper each cluster configuration will be created with only one particular instance type. A single instance type is used in line with previous research. A number of metrics such as throughput and latency will be used to measure the effectiveness of a range of different configurations. This will provide an ability to compare and contrast the performance and cost efficiencies of each configuration from both a scale-out and scale-up perspective.

Scaling-Out refers to horizontal scaling of a cluster consisting of a particular instance type. This horizontal scaling means increasing the number of instances in a cluster from X to Y in quantity. It is classified as scale-out if $X < Y$ and scale-in if the reverse is true. There is no change in the machine instance type, the only change is relative to the increase or decrease in the number of the instances deployed in line with the workload.

Scaling-up is the process of scaling from an instance type Instance-1 to another type Instance-2. Similar to the scale-out, scale-up occurs if Instance-1 is less powerful than the newer Instance-2 and scale-down is when the reverse is true. Scaling up or down is also known as vertical scaling in the traditional sense of parallel computing using multiprocessor or multicomputer systems (Hwang & Xu, 1998).

For simplicity and to ensure that the research is in line and directly comparable with previous research, the paper will adopt the ECU (Elastic Compute Unit) defined by AWS (Amazon Web Services, 2017) as a measure for evaluating a machine instances compute capacity (Hwang, et al., 2013). Further specific details

around the definition of an ECU are presented later. Amazon removed their ECU (Elastic Compute Unit) metrics in favour of the more traditional vCPU (virtual CPU) in 2014 (Gartner, 2014). The demand for the ease of comparison between the various instance types available through the use of ECU values prompted AWS to return the ECU values, alongside the vCPU values to their pricing and instance description (Amazon Web Services, 2017). These ECU values will be used for some of the metrics to follow.

As mentioned previously, the scope of this paper is to perform and gather benchmark results across multiple cloud configurations. Scale-up and scale-out deployments will be configured and the benchmark tools run against them to gather the appropriate metrics. Based on the comparisons between scale-out and scale-up carried out in previous research, the objective is to review similar instances using similar metrics. The additional knowledge gathered from these benchmark results will enable a more comprehensive assessment of the scale-up and scale-out configurations to be completed. The metrics include those which are highlighted as some of the most appropriate for comparison between the performance tests such as "response time", "requests per second" and "concurrent users" (Boonchieng, 2014). Additionally cloud comparison metrics such as, "speed-up", "elasticity" and "productivity" will also be explored (Hwang, et al., 2014). Further metrics concerning the potential for increasing performance through the use of additional threads from the benchmark application and the impact this has on the cloud instances performance (CPU activity level) are also gathered for comparison.

The metrics and formulae that will be used for analysis and comparisons in this paper are numbered below:

1) **Average Response Time (second)** (Boonchieng, 2014)

   This metric is gathered to ensure a comparison between the different benchmark runs takes into account the resulting increase or decrease in response times. Without taking this metric into account the implication to the application from the resulting change in test configuration would not be accurately considered.

$$Average\ response\ time = \frac{\sum Response\ time}{N}$$

   Response time refers to the time per second required to respond to the request.

   N indicates the number of responses.

2) **Request per Second (request/second)** (Boonchieng, 2014)

   Details the overall throughput / performance of the benchmark run. Provides a metric to enable the comparison of performance between different configurations and judge the positive or negative influence that changes have.

$$Request\ per\ second = \frac{\sum Requests}{T}$$

   Requests refers to the number of total requests.

   T indicates the time consumed during the data transmission process.

3) Thread count

Concurrent requests are the number of requests that an application can service concurrently.

Typically the limits of this can be tested in benchmarking through the use of threads as each thread would represent one application connection. Increasing the number of threads will generally increase the throughput of the application. This can positively affect the number of concurrent requests being made and thereby increase the performance of the application by reducing the time required to complete the operations (Tullsen, et al., 1995). Previous benchmarking results assumed the default thread count (single) when comparing and contrasting the performance and productivity of cloud deployments. Using a single thread does not represent real world multi-threaded applications it was determined that this would be a relevant metric to measure. This will be a metric that is manually adjusted and recorded in line with the benchmark performance results.

4) Speed up (Hwang, et al., 2016)

Speed gain of using multiple nodes. This formula is required to evaluate the benefit achieved when performing the operation through a larger AWS configuration.

Considering a cluster configuration Λ. T(1) is the execution time of a task on a 1-ECU instance whereas T(Λ) is the execution time of the same task on a virtual cluster Λ. Speed up is defined as:

$$Speedup\ (\Lambda) = {}^{T(1)}\Big/_{T(\Lambda)}$$

5) Efficiency (Hwang, et al., 2016)

This metric is calculated so as to provide a calculable and comparable score for the efficiency of a configuration. The score enables the amount of unused resources to be quantified based on the performance improvement the change in configuration brings.

Assume that the cluster is built with n instance types. The type-I has n_i instances, each with an ECU count c_i. We calculate the total cluster ECU count by:

$$N(\Lambda) = \sum_{i=1}^{i=n} n_i * c_i$$

This N(Λ) count sets a ceiling of the cluster speedup.

Now, we are ready to define the cloud efficiency for the cluster Λ in question as follows:

$$Efficiency\ (\Lambda) = Speedup(\Lambda)/\ N(\Lambda)$$

6) Productivity (Hwang, et al., 2016)

Similar to efficiency. The productivity metric provides a means with which to measure and quantify the overall performance of particular instances and configurations based on their cost.

Generally cloud productivity is contributed by the following three items, all related to the scaling factor.

1) System performance – could be measured as throughput in terms of transactions per second or the response time.

2) System availability as an indicator of Quality of Service (QoS) measured by percentage of uptime. (Typically close to 100%)

3) Cost for rented resources measured by price.

For example, let Λ be a cloud configuration in use. Cloud productivity has been defined by the following three factors, all being functions of Λ.

$$P(\Lambda) = \frac{p(\Lambda) * \omega(\Lambda)}{C(\Lambda)}$$

Where p(Λ) is a performance metric used (speed or throughput). The weight, w(Λ) is a measure of the Quality of Service (QoS) of the particular cloud. To simplify the QoS measure, it has been observed that the QoS can be approximated to the service availability measure. According to CloudSquare on hundreds of cloud services surveyed, over 90% of them have 99% or higher availability (CloudSquare, 2017). The C(Λ) is the user cost to rent resources to form the virtual cluster Λ.

3.2 Testbed Configuration

3.2.1 Amazon Web Services (AWS) & Amazon Elastic Compute Cloud (EC2)

Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality. To provide a robust and highly available system, AWS is split into geographically diversified regions, each region comprising of multiple smaller geographic areas called availability zones. There are many different consumable products and services within AWS, the most relevant to the work being undertaken in this thesis being the Amazon Elastic Compute Cloud (EC2). Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing capacity to quickly scale both up and down, as requirements change. Amazon EC2 changes the economics of computing by allowing consumers to pay only for capacity that is used. There are a number of reasons why AWS EC2 was chosen as the IaaS platform for this research. The primary reason was to ensure that the research done would use the same environment as previous research in order to facilitate a comparative analysis (Hwang, et al., 2016). Additionally AWS was chosen as it dominates the public cloud (IaaS) usage for enterprises globally. The RightScale survey depicted in figure 7 below shows that 57% of respondents adopt AWS. Enterprise adoption of AWS grew from 50 percent to 59 percent between 2015 and 2017 reports (RightScale, 2017).

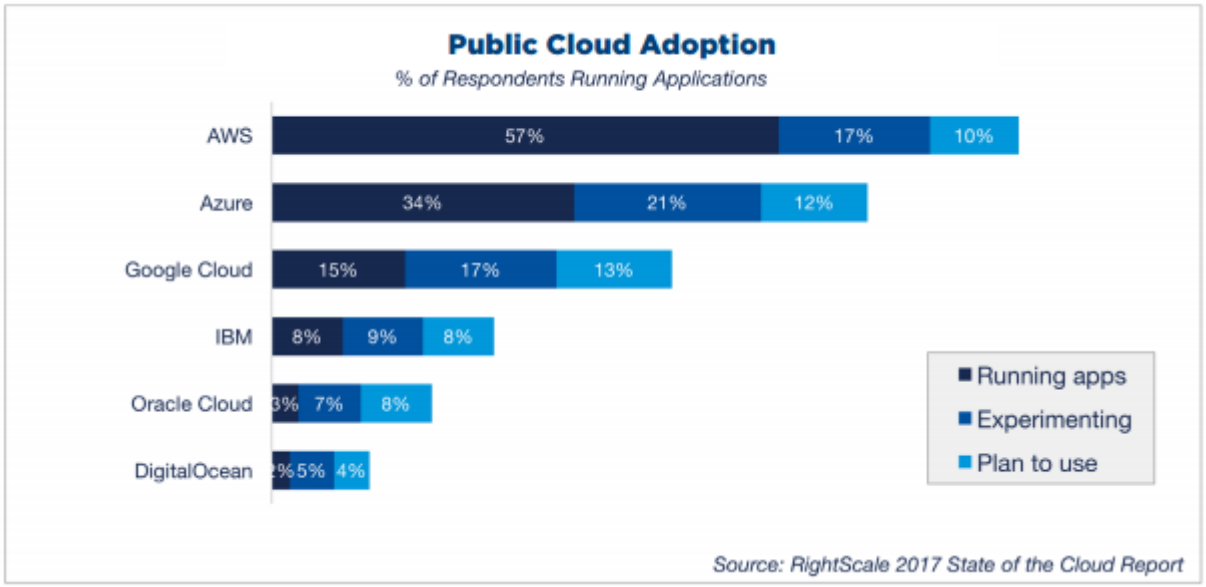

Figure 7: Public Cloud Adoption – (RightScale, 2017)

3.2.1.1 AWS EC2 Configuration Options

As highlighted, AWS EC2 was chosen to provide a similar basis of technology to build on the previous work done regarding scale-up and scale-out benchmarking in the cloud (Hwang, et al., 2016). Since the previous research was done, Amazon has removed a number of the instances used from the available catalogue. This

meant that to replicate the tests as closely as possible those instances previously used and remaining available for selection were used. The three repeatable instances (m3.medium, m3.xlarge and c3.xlarge) were chosen along with a 4th, memory intensive offering (r3.large); details of each and their particular characteristics are available in table 1 below. The memory rich AWS instance type "r3" was chosen as the addition as it is recommended as the best overall instance for MongoDB instances due to the balance of compute and memory (MongoDB, 2017) (ScaleGrid, 2014). None of the previously used smaller offerings (m3.small) were available. This similarly small sized instance was required in order to have a method of gathering statistics on a standardised 1 EC2 Compute Unit (ECU) for comparisons. In trying to identify 1 ECU instance, only AWS instances of type "t2" were close. However, at the time of research these smallest "t2" type instances were only available with "burstable" compute. Due to the expected short-term usage of instances for benchmarking runs and unknown allocation of "burst credits" for the compute aspects it makes them a less predictable instance to consume. This unpredictability means that calculations cannot be accurate and ruled them out of consideration for addition in this research.

As mentioned previously, Amazon AWS defined EC2 *Compute Unit* (ECU) as an abstract unit to quantify the computing capacity of each instance type. Memory and storage also impact the ECU count due to the additional performance they would provide to the instance. By the 2009 standard, the performance of a 1 ECU instance is roughly equivalent to the CPU capacity of a 1.2 GHz 2007 Xeon processor (Hwang, et al., 2014). ECUs are still used in this research as a means of having an understanding of the expected capabilities of the instances and clusters. Their usage also means that the cloud metric formulae previously defined can be used to enable accurate comparisons and conclusions to be drawn. A table of the instance types and their specifications are contained in the table below, this information was gathered from the AWS on-demand pricing and is based on the AWS West (Ireland) options and pricing as of April 4th 2017(Amazon Web Services, 2017).

Instance Type	Processor (Intel Xeon)	ECU	vCPU (Cores)	Memory (GB)	SSD Storage (GB)	Price ($/hour)
m3.medium	E5-2670 v2	3	1	3.75	1 ×4	0.073
r3.large	E5-2670 v2	6.5	2	15	1 x 32	0.185
m3.xlarge	E5-2670 v2	13	4	15	2 x 40	0.293
c3.xlarge	E5-2680 v2	14	4	7.5	2 × 40	0.239

Table 1 - Machine Instance Types used in YCSB tested MongoDB scale-up experiments on Amazon EC2

Instance Type	Use Case	Processor (Intel Xeon)	ECU	vCPU (Cores)	Memory (GB)	SSD Storage (GB)	Price ($/hour)

t2.micro	Mongo Config Server	Intel Xeon	Variable	1	0.5	Elastic Block Storage	0
m3.medium	Mongo Shard Instance	Intel Xeon E5-2670 v2	3	1	3.75	1 ×4	0.073
c3.xlarge	YCSB Benchmark	Intel Xeon E5-2680 v2	14	4	7.5	2 × 40	0.239
c4.4xlarge	Mongo Shard Router	Intel Xeon E5-2666 v3	62	16	30	Elastic Block Storage	0.905

When scaling-out for the purpose of this research there are additional instances created in AWS and some application configuration is required to incorporate these new instances into the cluster. The scale-out of the cluster will be done with a fixed instance type in this paper, in line with previous research. On the other hand, for the scale-up experiments, the instance type under test is cycled through the table by creating the appropriate sized instance in AWS. Manual scaling is applied in all experiments. Auto-scaling is not applied in scaling experiments on EC2 due to its inefficient and brute force provisioning policy. Mixed instance scaling (having instance types: m3.large and c3.large in one cluster for example) will also not be used due to the more difficult and inefficient means of scaling the cluster (Hwang, et al., 2014) (Hwang, et al., 2016). Due to the different configuration requirements of scale-out, there are some additional instances which need to be deployed. The specific usage of each is detailed below and the exact reasons for the configuration will be outlined in the scale-out section to follow. Table 2 below highlights the details of each of these instances.

3.2.2 Load Simulation using MongoDB

MongoDB is used in this project to benchmark the various workloads. MongoDB is a NoSQL Database technology (MongoDB, 2017). NoSQL databases provide a means of storing and retrieving data which is modelled, in a format which is not the typical tabular format seen in relational databases such as Oracle and SQL. They have become increasingly popular in the last number of years due to the requirement from web-scale companies to solve scalability and big-data performance issues that relational databases were not designed to address (Mohan, 2013). NoSQL provides a means of accessing and analyzing large amounts of unstructured or semi-structured data. There are many different types of data structures used by NoSQL databases (e.g. key-value, wide column, graph, or document). These are significantly different to the

Table 2: Machine Instance Types used in YCSB tested MongoDB scale-out benchmarking on Amazon EC2

structures implemented in relational databases typically. The different NoSQL databases (Cassandra, MongoDB, Couchbase etc.) all offer different capabilities and are chosen based on the problems that need to be solved. Choosing the appropriate NoSQL DB enables the relevant workloads operations to be processed faster in NoSQL.

Most NoSQL databases provide increased speed, availability and flexibility over traditional relational databases. The sacrifice for these benefits is typically consistency. NoSQL databases offer a concept of "eventual consistency". This means that nodes will “eventually” receive the required database updates (normally within milliseconds). Due to this eventual consistency there can be a concern that so some queries might not return data immediately or else return inaccurate / stale data.

MongoDB provides recoverability though the use of replica-sets (MongoDB, 2017). These are groups of processes which maintain the same data set, thus enabling redundancy and high availability. A replica set contains several data bearing nodes, one of which is deemed the primary. The configuration of one replica set can be seen in the figure 8 below. All write operations go through the primary node as it records all changes to the data sets in its operation log. Secondary’s apply operations from the primary asynchronously. Read operations default to go through the primary also but a read preference can redirect the reads through secondary’s also.

The purpose of the secondary instances and the copy of the data sets is to provide continued access and functionality despite the failure of one or more members. When an instance failure occurs, an election of a new primary will take place and the new primary will then take over read and write operations per figure 9

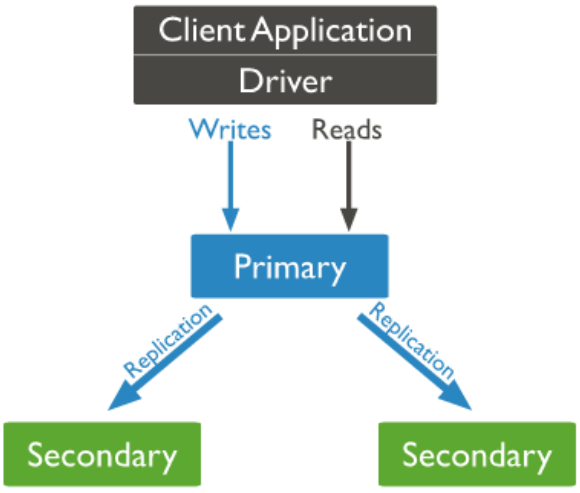

Figure 8 - MongoDB Replica set – MongoDB.com

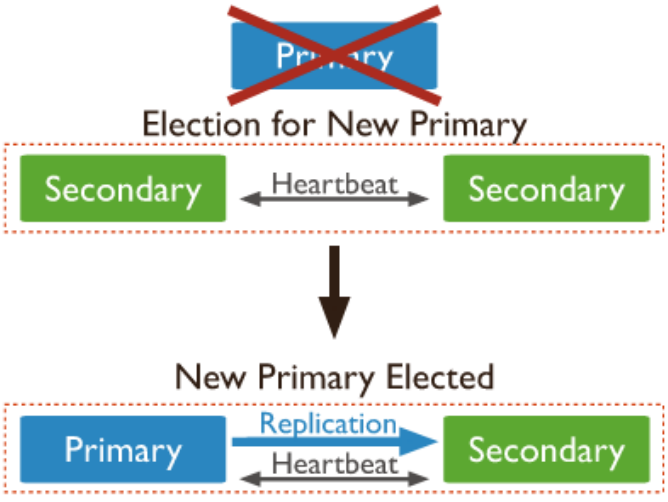

Figure 9 - MongoDB Instance Failure / Election - MongoDB.com

below.

MongoDB is being used as the relevant application in this empirical evaluation in order to facilitate the benchmark simulation of different loads. Additionally, the ever increasing usage of NoSQL databases makes an investigation into their performance and productivity metrics relevant. The performance characteristics of MongoDB deployments can be benchmarked using some well-known and commonly used tools. This enables a large number of options for tests and comparison. Apart from these, the primary reason it has been chosen is because MongoDB is designed for both a single instance vertical scale-up and multi-instance horizontal scale-

out environments (Chodorow, 2013) and so will enable accurate comparisons to be made for the purpose of this research.

Scaling up is typically the easiest option to achieve using MongoDB as it just involves changing the instance sizes, however its limit is much more quickly reached, there is only so much RAM / CPU that can be contained within one instance at a reasonable cost. After reaching a certain limit of instance capability, the performance gains achieved by continuing to scale-up can be reduced by the exponential cost increases. An effective alternative can therefore be to leverage a scale-out deployment.

Scale-out enables the performance or storage to be expanded to an almost limitless capabilities by leveraging a cluster with sufficient capabilities and adding instances as and when required. It is cheaper and more scalable typically but is more difficult to administer (Chodorow, 2013). A feature known as sharding is used to provide horizontal scale out capabilities for MongoDB.

Sharding is the method used by applications to distribute data across multiple compute instances. Each shard contains a subset of the sharded data. Each shard is basically an independent database, all of the shards collectively then form one logical database as can be seen in figure 10 below. The overall configuration of the required instances is shown in figure 11 below. As is highlighted, the data is accessed through a mongos instance, otherwise known as a Shard Router. This Router routes the reads and writes from applications to the collections underlying shards. The other requirement in a sharded deployment is a configuration (config) server. Configuration servers store metadata about the cluster in the configuration database. The mongos router instances cache this configuration data and use it to quickly route reads and writes to shards. The read and write workloads are distributed across the shards in a cluster as shown for collection 1 in figure 12 below. Each shard can then process a subset of the clusters overall operations. This allows both read and write workloads to be scaled horizontally across the cluster by adding more shards (MongoDB, 2017). This scale-out strategy means that the performance of any further read and write queries can be increased as the documents can be read simultaneously across many servers instead of from a single large server. MongoDB will balance the data among servers, even if they are added on the fly and will also determine where data is written and read from, thereby not putting any additional overhead on the application.

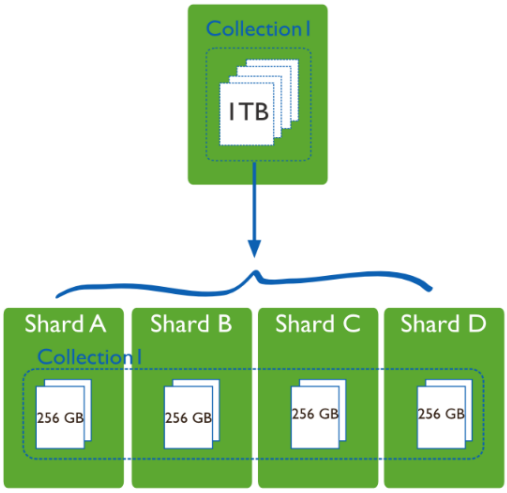

Figure 10: MongoDB Sharding

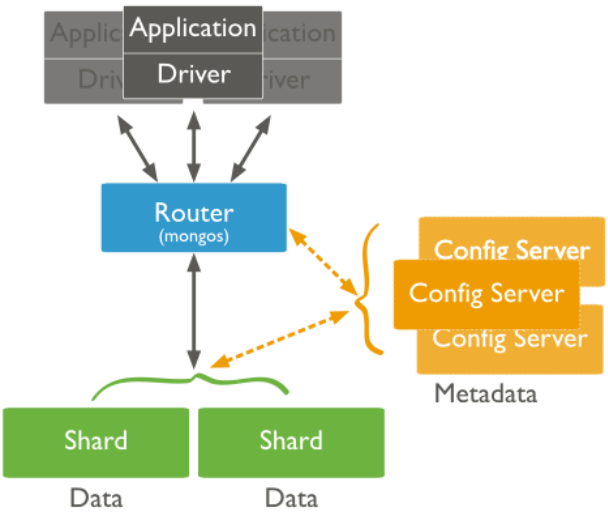

Figure 11: Sharded cluster configuration - MongoDB.com

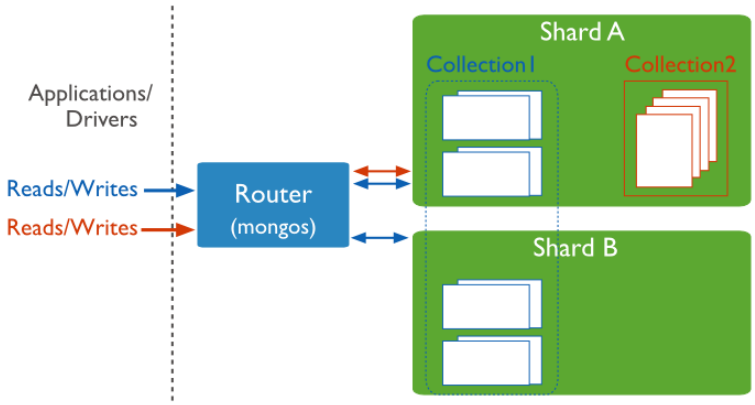

Figure 12: MongoDB Sharded cluster data flow - MongoDB.com

3.2.3 YCSB Benchmarking

YCSB – Yahoo! Cloud Serving Benchmark is a popular benchmarking tool and a common framework which was primarily developed by Yahoo! Co. to test their PNUTS datastore (Cooper, et al., 2010). YCSB has since been open-sourced and is available from GitHub (Cooper, 2017). The tool consists of an extendable workload generator and a set of core workloads which it can execute. It enables comparative performance of various NoSQL databases. This ability to test different databases is due to the tools adaptability and the fact that developers can create different interfaces relevant to the specific database. There are different workloads which are used to provide a relevant analysis of the various NoSQL databases. The goal as defined by the creators was to "*develop a framework and common set of workloads for evaluating the performance of different "key-value" and "cloud" serving stores"*. The type of large scale web type applications which can be replicated in the benchmarking with YCSB would typically consist of a very large number of small queries. The extensibility of the tool enables new workloads to be developed to examine a specific system or aspects of an application so that it can be used to benchmark new cloud database systems (Yahoo!, 2010).

YCSB is the benchmarking tool of choice in this research as it is a proven benchmarking tool for NoSQL and MongoDB in particular (Floratou, et al., 2012) (Kashyap, et al., 2013). Its use has been detailed and documented in multiple examples online to provide an overview of the capabilities and enable comparison between different NoSQL databases (MongoDB, 2015) (ScaleGrid, 2015) (United Software Associates, n.d.). It provides a method of building on the research previously carried out by using the same tool as used in the previously documented research (Hwang, et al., 2016). YCSB also provides the functionality required to gather additional metrics around the usage of multiple concurrent threads.

The architecture of YCSB can be summarised as follows; The YCSB client is a java program whose purpose is to generate data, which can be loaded to the particular database to be tested. It is also able to perform operations to create specific workloads. The standard workloads include those shown in *Table 3* below and also provide the ability for users to create their own specific one (Cooper, 2010).

Core Workload	Description	Details
A	Update heavy workload	Mix of 50/50 reads/writes.
B	Read mostly workload	95/5 read/write mix.
C	Read Only	100% Read
D	Read latest workload	New records are inserted, and the most recently inserted records are the most popular.
E	Short Ranges	Short ranges of records are queried, instead of individual records.
F	Read-modify-write	The client will read a record, modify it, and write back the changes.

Table 3: YCSB Workload Types - https://github.com/brianfrankcooper/YCSB/wiki/Core-Workloads

The overall architecture of YCSB is shown in figure 13 below. The workload executor, which can be seen in the middle, manages multiple client threads. Each of these threads completes a specific number and type of

operations to the database interface layer. The operations are completed according to the workload and are done at two times. Initially operations are issued when the database is being loaded (load phase). Once the load is completed the operations are run during the workload execution (transaction phase). The rate that the requests are generated is throttled by the threads. The number of threads and other items such as number of records to insert, number of operations to perform etc. are either entered in the command line or through the use of a parameter file. These are shown at the top of the figure as feeding into the client. These configuration items enable control of the workload against the test database. The latency and throughput of the operations is gathered and issued to the statistics module. When the workload completes, the measurements are aggregated and the average, 95th and 99th percentile latencies, and either a histogram or time series of them is presented back through either the command line or a specified output file (Cooper, et al., 2010).

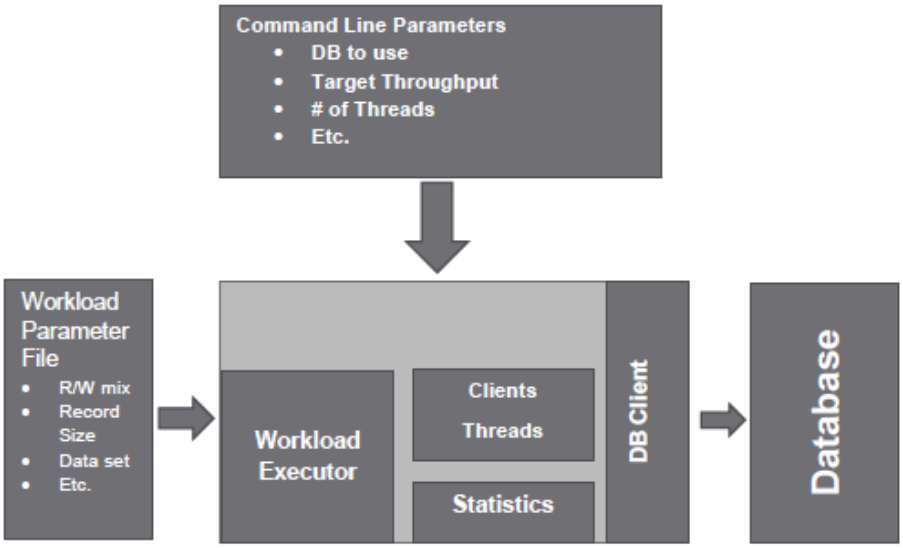

Figure 13: YCSB Architecture Diagram – Cooper et al., 2010

4 Empirical Results

4.1 Scale-Up overview

4.1.1 Configuration

For the scale-up benchmarking test to be completed a number of items needed to be configured. Firstly the 4 different sized instances needed to be created on AWS. Each instance was configured with Ubuntu 16.04 LTS. The overall configuration is shown in figure 14 below.

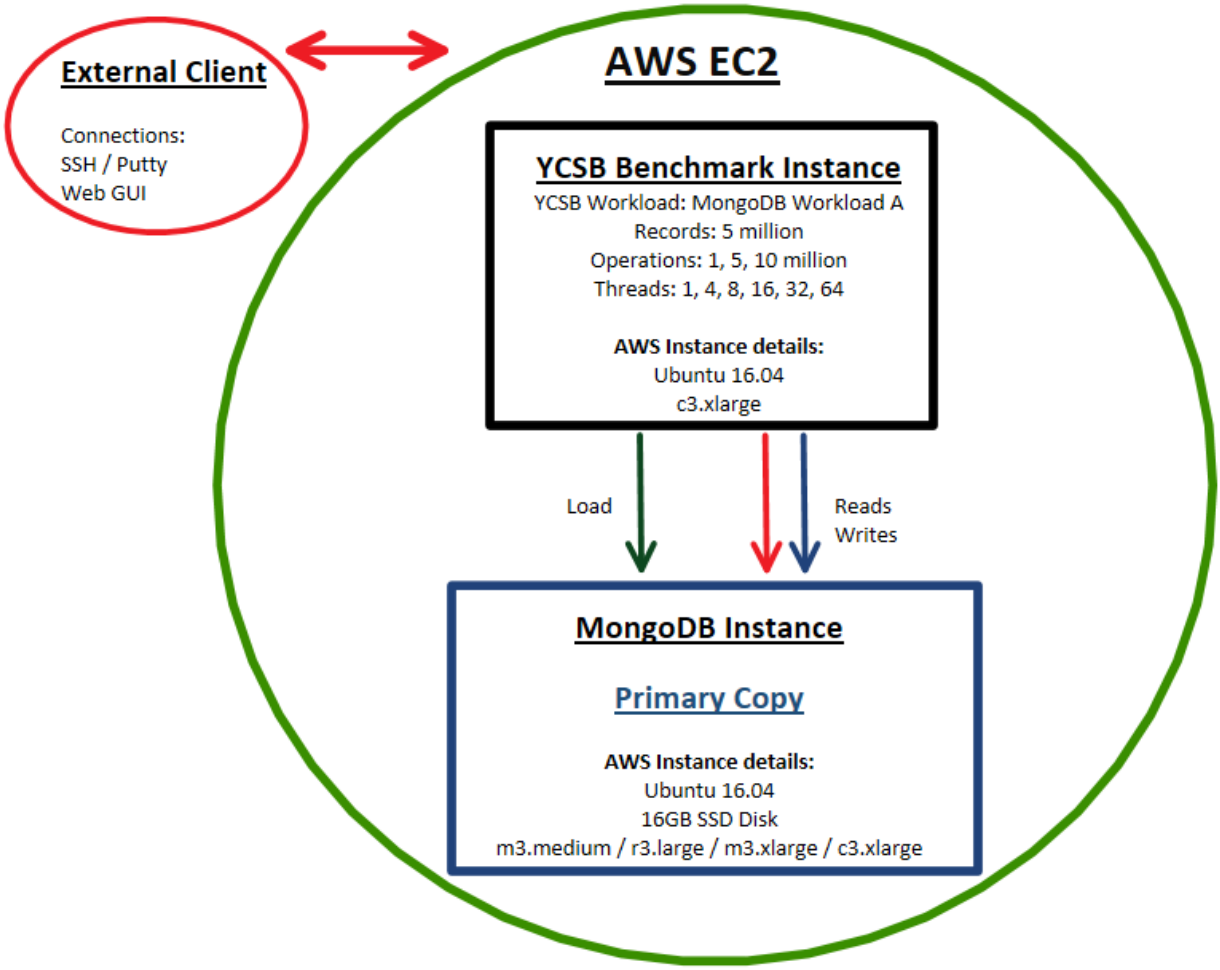

Figure 14: Scale-up instance configuration overview

After each Ubuntu instance was initialised it was then required to configure MongoDB on each. Due to the number of records being inserted, each of the scale-up instances needed to be configured with 16GB of SSD disk space to provide sufficient capacity.

Each scale-up AWS EC2 instance had MongoDB installed as per MongoDB guidelines and recommended procedures (MongoDB, 2017). There was no creation of configuration servers or replica sets for any of the scale-up instances. Replica sets were not configured as it is not a requirement for a functional system for the

purposes of benchmarking. In reality production data would be protected against failure by the use of replica sets. As the data replication is done asynchronously it would not have affected the overall performance of the primary nodes ability to ingest data. Also as the instances were configured in the same manner, the results will not have been affected. The same MongoDB configuration was deployed for each of the instances.

For benchmarking, YCSB was used as the basis for all tests. 5,000,000 records were inserted into the MongoDB database prior to running each workload. The same number of records and record size was used to provide a common baseline. The number of operations and number of threads was increased in line with the planned testing. The number of operations was increased between 1, 5 and 10 Million when benchmarking each instance to provide an accurate impression of the different responses during the tests. The number of threads was increased through the values, 1, 4, 8, 16, 32 and 64 so as to investigate the maximum throughput being reached without sacrificing latency. There were multiple independent runs performed for each configuration. To gather the results from YCSB average throughput, and latencies for reads and updates as well as overall run time and operations per second were calculated.

The YCSB setup consisted of one client server and was installed on Ubunto 16.04 from GitHub as per best practices (Cooper, 2017). This separation of the YCSB client from the MongoDB instance was to ensure the YCSB client was not competing with the database for resources.

Initially the YCSB client instance was based on an r3.large instance on AWS. After running the initial benchmark tests with 1 thread from YCSB this appeared to perform adequately and without any bottlenecks on the part of the benchmark instance observed. However, when further tests were done with an increasing number of threads from the YCSB tool it was observed that the r3.large instance was not compute intensive enough. As the threads were increased in the YCSB application there was additional load spread across the 2 vCPUs available on the r3.large instance. As the threads increased beyond 8 it became apparent that this additional load being placed on the YCSB instances vCPUs was too much for it to handle adequately. This resulted in the vCPUs on the benchmark instance being overloaded and unable to issue the workload operations quickly enough. This meant that the MongoDB instance under test was not being tested to its full potential and so the results were inaccurate. This issue was not apparent when reviewing the results initially and resulted in similar performance results being seen across the different instance types. For this reason an instance with more compute resources was deployed as the base for the YCSB benchmark application. This was the instance type c3.xlarge which has 4 vCPU and an ECU score of 14. This instance was chosen over smaller or cheaper instance versions to ensure that the performance benchmarking results would not be affected by the capabilities of the YCSB client. The CPU activity levels were monitored during further testing and did not reach 100%. YCSB “Workload A” was chosen as that best represents a significant write workload (50/50 Read / Write) and gives a realistic view as to the potential capabilities of MongoDB in both reads and writes. As stated previously, any workload could be chosen and custom workloads can be designed with the

relevant application in mind however testing many different configurations is outside of the remit of this thesis.

4.1.2 Results observed

The investigations and results concluded can be mapped out as follows. The results are split into two definitive sections, the first detailing the results for all of the research done on scale-up and the second for the scale-out deployment. In each section there is initially a comparison between each deployments (different sized for scale-up vs different number for scale-out) handling of increasing workloads with 1 YCSB thread is completed. This is done using each of the metrics such as throughput, latency, efficiency and productivity where appropriate. The next results documented provide a comparison between these metrics when gathered for the deployments when considering increasing the number of threads from the YCSB benchmark tool through 1, 4, 8, 16, 32 and 64. The results of the monitoring of the CPU and memory usage of each deployment for the increasing workloads is also presented.

The scale-up deployment and benchmarking was carried out first. The main reason for this was to ensure that the deployment was successful and that data being gathered provided for usable results. The results of the scale-up deployment and benchmarking are shown in tables to follow.

As previously outlined, the benchmarking was performed with three different values of total operations to be carried out. The purpose of this was to determine if there was any significant increase or decrease in the overall performance of the system under test when there was a large increase in the operations to be performed. When performing initial testing, the three different values of operations (0.1, 1, 5 million) used with YCSB in the previous research paper completed relatively quickly. The time which the 0.1 million operations took to complete in was so short it was decided that a longer timeframe would be more accurate and assist with gathering metrics on larger workloads. The 0.1 million operations was therefore removed as a figure and the number of operations used was 1, 5 and 10 million.

Even though both average read and update latencies were gathered from the results, to produce a meaningful graph update latency was chosen as the metric to represent latencies. There was minimal difference observed between both latencies for most tests when reviewing the results. I decided to take the average write latency as the metric as it was observed that it would normally be slower. This would be expected due to the additional latency associated with writing data to disks and the underlying penalties associated with protecting the data as it is being written. Reads would not incur the same performance impact as the data is only being retrieved. Due to this the latency associated with writes is more impactful to applications typically.

4.1.2.1 Scale-up: Observations on increasing workloads (1 – 5 – 10 Million operations)

As can be seen from figure 15 below, increasing the number of operations between 1, 5 and 10 million has little impact on the overall throughput being achieved. This graph is representative of the testing done with the

default single thread from YCSB specified similar to the previous research. These results are in contrast to the previously documented YCSB benchmarking (Hwang, et al., 2016), where there was a more noticeable difference between the increasing numbers of operations. This difference appears to be related to the fact that there is a different application (HBASE) under test. The previous research identified that the operations for HBASE are memory intensive and so would potentially benefit from more of the data being in memory (Hwang, et al., 2016). As the YCSB workloads are dependent on the application being tested (MongoDB / HBASE etc.) then additional benefits may accrue when continuously performing writes to the application. To further clarify the difference observed between these observations would require additional investigation and significant time which was outside of the scope of this project. In the testing done here, the small difference between the workloads shows that there is no additional advantage to having larger numbers of operations taking place. What can be observed is that that throughput does not increase in a linear fashion through the instances. The increase is the most substantial for both the r3.large and c3.xlarge instances. The r3.large instance has the largest amount of configured memory, and the c3.xlarge has the largest amount of processing power. Of those tested, it would be expected that the r3 instance, having the largest configured memory and being on the recommended MongoDB instance type (MongoDB, 2017) would perform best with 1 thread. The m3.medium instance would always be expected to perform worst due to it having the smallest CPU and memory configuration. What is unexpected is the difference between the m3.xlarge instances and the r3 and c3 instances. The m3.xlarge instance has a similar memory configuration and ECU score as the r3 and c3 instances respectively so the exact reasons for the consistently poorer results are not initially clear and would warrant further investigation.

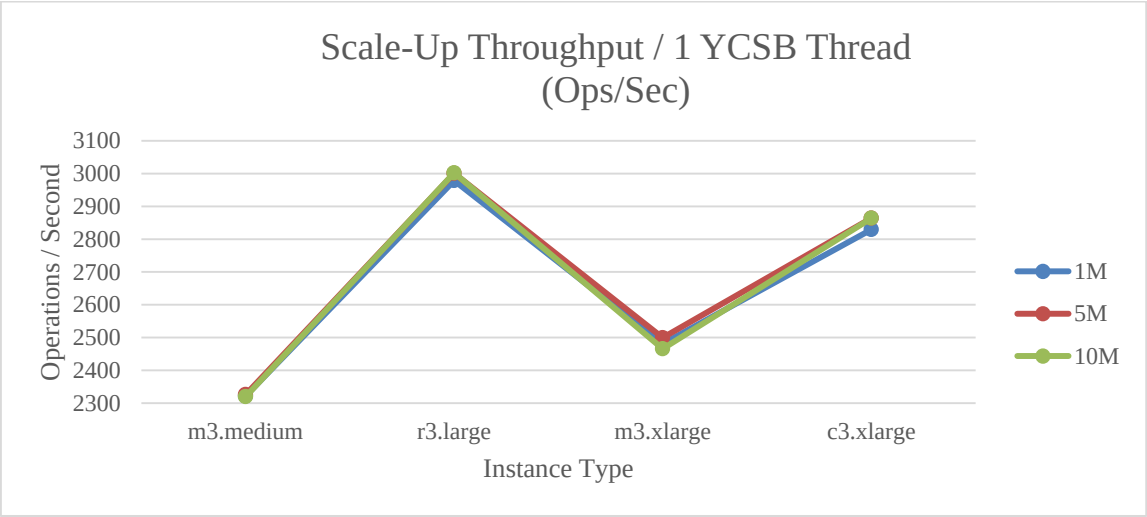

Figure 15 : Scale-up throughput per instance (1 thread)

Figure 16 below shows the difference between the latency of the differing number of operations on each instance. In a trend similar to that observed in the throughputs shown above, there is a very limited difference between the averages for each metric of 1, 5 and 10 million operations. Likewise, the latency decreases in line with the increase in throughput observed in the previous table. This again shows that the instances with the best compute, c3.xlarge, and the most memory, r3.large provide the best result for the application. The

m3.xlarge instance is the most expensive, yet not the most appropriate for this applications benchmarking it appears.

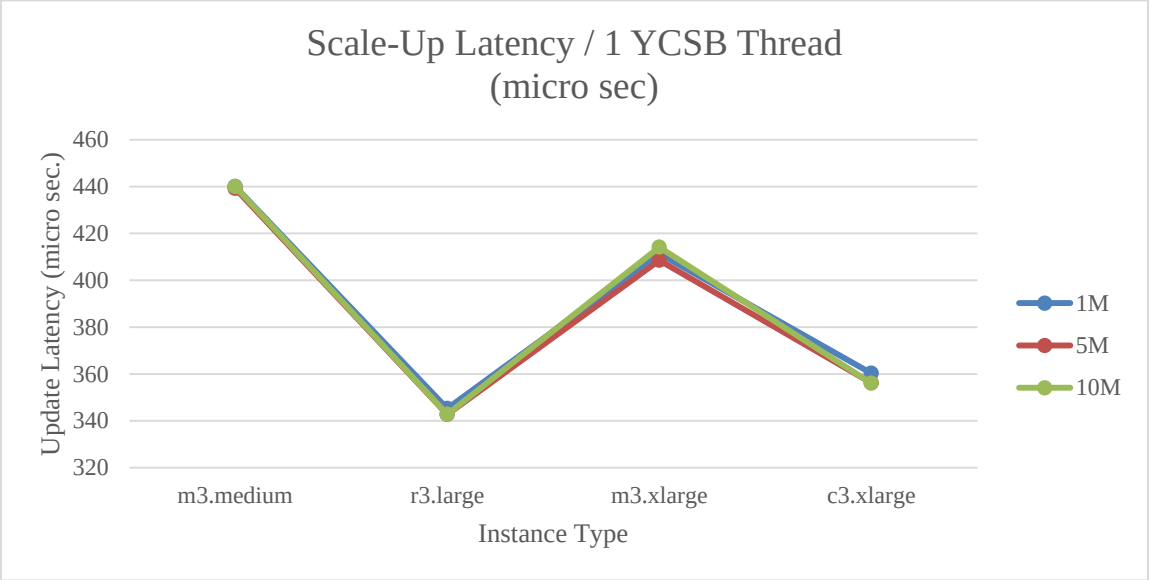

Figure 16: Scale-up avg. update latency (1 thread)

The previous metrics (throughput and latency) were gathered directly from the YCSB results themselves. Figure 17 graphs the calculation of the efficiency (formula 5). The efficiency calculation requires the use of the speedup formula (formula 4) to compare the time taken by the instance to complete the operations in comparison to the time taken by a 1 ECU instance. As there was no guaranteed 1 ECU instance available for use, the smallest configurable option was the m3.medium instance. The results from this instances tests were used to calculate a representation of a 1 ECU instance as close as possible. To calculate the time for a representative 1 ECU instance the m3.mediums results were divided by its ECU count. The m3.medium instance has an ECU count of 3, which meant that the total time for m3.medium to complete the operations was divided by 3 to provide the 1 ECU representative time.

This workaround means that for the m3.medium instance its speedup and efficiency metrics would therefore be incorrect. As the 1 ECU time was created by dividing the m3.medium instances times by 3, the use of this figure in the speedup formula would mean that m3.medium would show as exactly 3 times faster. This metric would then mean that it would incorrectly show as 100% efficient. This error was accepted as a means to compare the more relevant and higher throughput instances. As the productivity formula (formula 6) was not reliant on any of these incorrect calculations the research would provide sufficient comparisons to enable an accurate conclusion.

The calculations enabling comparisons of the 1, 5 and 10 million operation results were done using the metrics gathered during the 1 YCSB thread benchmark run.

As can be seen from the efficiency graph in figure 17 below, all of the calculations carried out on the metrics from the increasing number of operations provide for the same results. The efficiency metric is based off the overall throughput and as previously identified, investigation into the unexpected similarities requires investigation outside of the scope of this paper. They are so close in the line graph that the column chart in figure 18 is provided to show more granularity. As previously outlined, the lack of an AWS EC2 instance with predictable 1 ECU means that to calculate the "Efficiency" of the other instances it is necessary to measure them relative to the performance of the m3.medium instance. The calculated "Efficiency" of m3.medium is 1 based on the method of calculating the formula. The efficiency of the other instances remains almost the same as the number of operations increases due to the minimal change in operations per second previously observed. The linear increase in time to completion across the instances when the number of operations increases means that the results are almost identical. As detailed in the formula section, the efficiency formula (formula 5) relies on a calculation of the speedup (formula 4) attained. The speedup is based on the difference in time to completion between the m3.medium and the other instances and so as the number of operations increased, so too did the time to complete resulting in the same overall result. The most interesting aspect which can be determined from the graph is that the r3.large instance provides a significant increase in efficiency in comparison to the other two large instances of m3.xlarge and c3.xlarge. This highlights that the increase in ECU does not translate linearly to an increase in efficiency. The fact that the r3.large instance provides more of an increase in performance for a smaller corresponding increase in ECU count is obvious and these results confirm that. The resources that each benchmarked application require could impact the efficiency with which the instance is used. In this case the r3.large instance provides the resources which can be most efficiently used by the 1 threaded YCSB MongoDB Workload A benchmark.

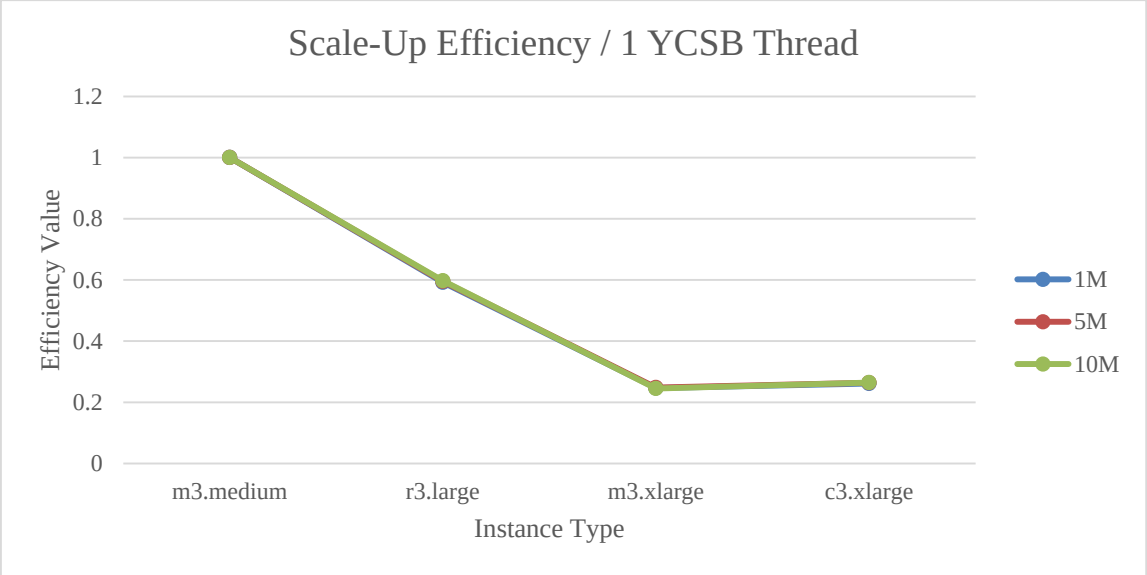

Figure 17: Scale-up efficiency (1 Million ops)

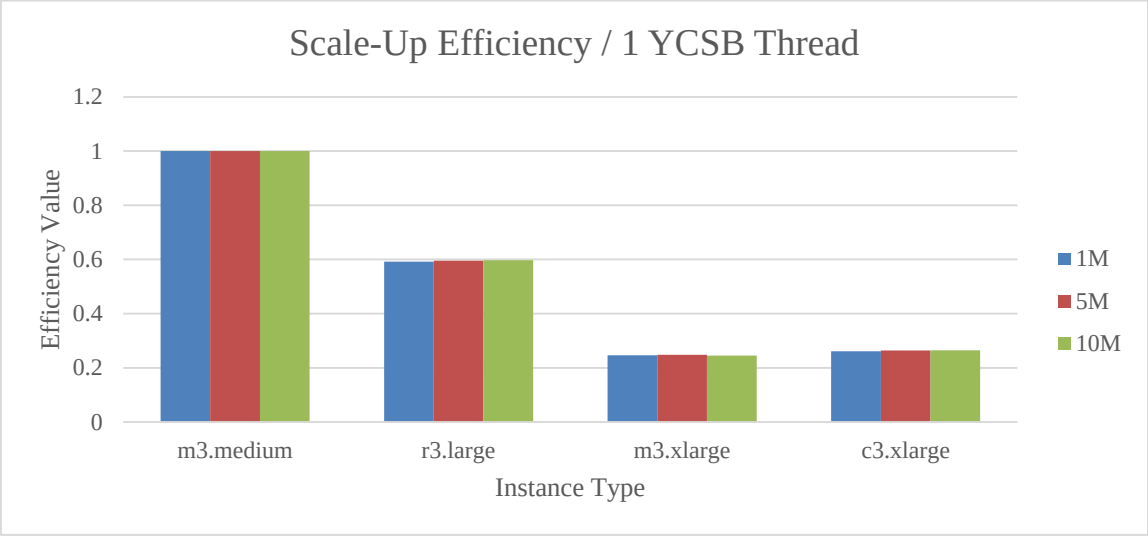

Figure 18: Scale-up efficiency (1 Million ops) - Column Chart

The productivity (formula 6) was calculated for each instance using 1 million operations, 1 thread metrics for the same reason as mentioned previously. In the productivity formula is a value for the cloud service rating. As AWS has a cloud service rating in the 99.9999% range (CloudSquare, 2017), the weight applied was rounded to 1. As can be seen in the line graph and column chart in figures 19 and 20 below, the increasing operations workloads made negligible impact to the overall productivity calculation. This provides a similar result to the efficiency graph above for the same reasons, similarity in throughput for the increasing operations count. The productivity calculation takes into account the number of operations completed and the cost of the instance. This calculation therefore will assist in highlighting the instance type which is providing best potential throughput for each dollar spent. The cheapest instance, m3.medium has the best productivity in the metrics generated with 1 thread being used from YCSB. The second most productive is the r3.large instance which would highlight that to reach highest productivity with memory intensive MongoDB workloads it is most appropriate to use a low ECU but high memory instance.

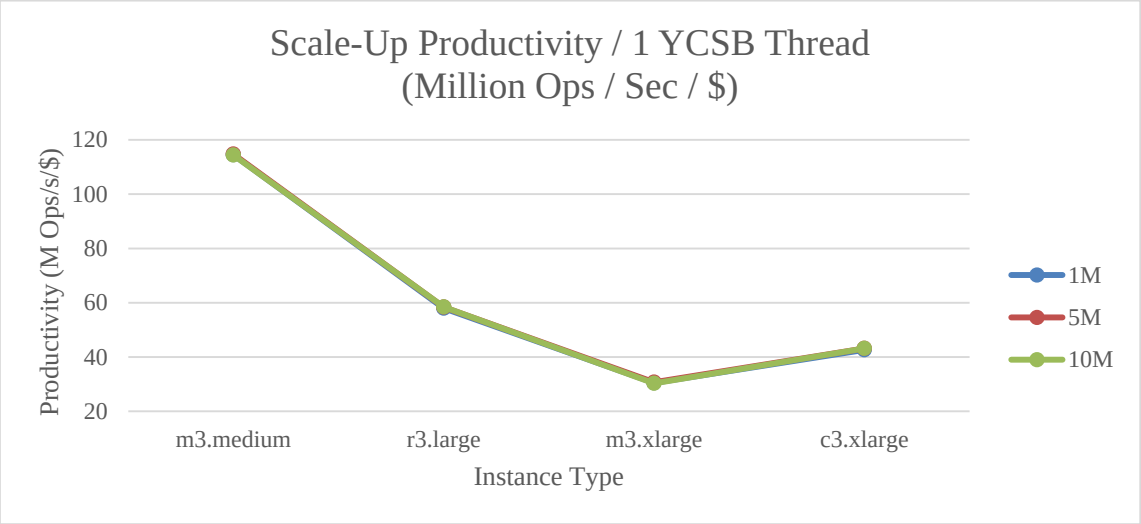

Figure 19: Productivity per instance (Million operations / second / $) - 1 YCSB thread

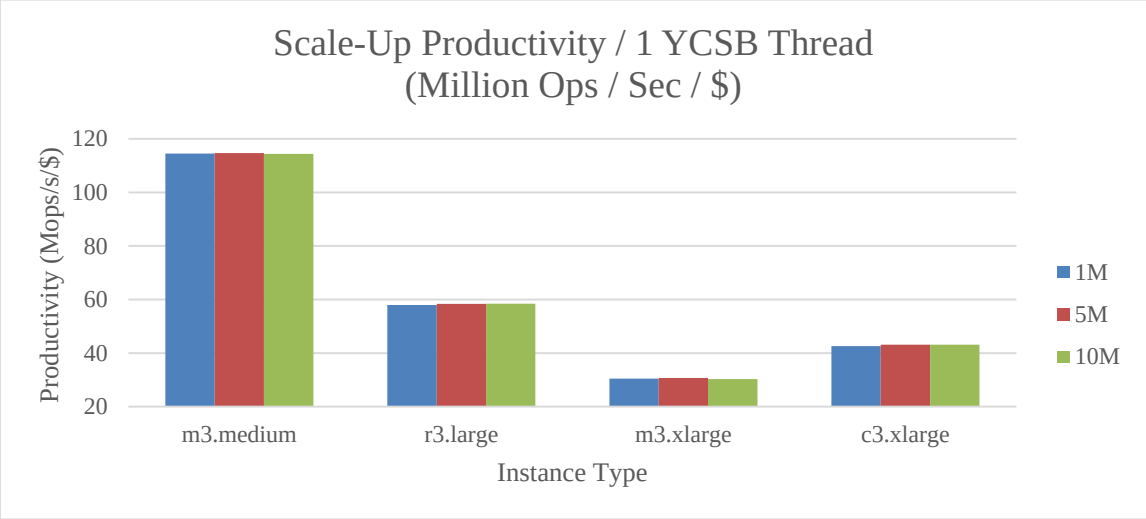

Figure 20: Productivity per instance (Million operations / second / $) - 1 YCSB thread – Column Chart

4.1.2.2 Scale-up: Observations on increasing YCSB thread count

As mentioned at the outset, this research paper would additionally investigate the impact that running multiple threads from the benchmark application would have on the overall performance and resulting metrics. The efficiency and productivity results in figures 17 and 19 above suggest that the m3.medium instance would be the best instance to use. The productivity figure is the one which would be most convincing to cloud consumers looking to implement a MongoDB deployment at first glance as it provides a metric which directly outlines the number of operations or performance which can be provided relative to the cost. As identified previously, the performance and cost are the main measurable factors influencing cloud adoption. To ensure that all considerations were investigated fully all aspects influencing the performance of the benchmark application need to be reviewed and scrutinised. The key parameter which YCSB provides to enable the performance investigation is the number of threads it can run. Running multi-threaded benchmark tests is therefore key to determining if the performance could be increased and latency decreased respectively. The

calculations were done using the metrics gathered from the 1 million operations, but varying numbers of threads for each benchmark run. The results of one set of operation values, e.g. 1 million, was done to provide an easier to interpret graph. The similarity between the 1, 5 and 10 million operation runs meant that there would not be any obvious impact from choosing one over the other.

As can be seen in figure 21 below there is a differing increase in throughput for all instances when the number of threads from the YCSB benchmark tool are increased. The m3.medium instance has the smallest increase, going from ~2950 Ops/Sec to almost double at ~4950 Ops/Sec before there is no further increase observed. The other instances all recorded more substantial increases with the relative increase in thread counts as can be seen below. The starting point from where we have generated the preceding graphs is 1 thread on the bottom left. As we have previously observed there is a slight difference between the throughputs for each instance with 1 thread. Increasing the threads from the YCSB benchmark instance significantly changes the observed throughput from each instance.

The results show that the performance of the instances is comparable to their relative ECU count highlighted in table 1. The instances with the largest CPU count and overall AWS ECU score perform the best. The performance observed shows that as the threads increase there is a certain point at which the instances capabilities to perform more operations peaks. After the instances relative peak is reached the performance drops off. This would be due to the increasing time that the workload operations are queueing due to there being insufficient processing capabilities in the relevant test instances. For example, the m3.medium instance with its 1 vCPU does not perform any better after reaching its peak of 4 threads.

The latency results observed and presented in figures 22 and 23 below should be viewed in conjunction with the performance graph in figure 21. Figure 22 and 23 are generated from the same results, they are just presented in different formats to show the similarity in latency between some instances. These metrics show that, the throughput continued to increase with thread count but observed peaks at 32 threads for m3.xlarge and c3.xlarge instances. However the operations per second only show aspect important to application performance. The increasing latency would have a negative impact on the application. When the increasing throughput is simultaneously viewed with the relevant increasing latencies it could be determined that the peak throughput may not produce the best application performance. This means that the application would potentially receive better performance with a reduced thread count, reducing the throughput but simultaneously reducing the latency impact. The exact throughput and latency that would be appropriate for different applications would depend on the specific applications requirements and best practice recommendations. In the tests conducted, in all but the smallest m3.medium instance where the latency increases with 4 threads, there is a significant increase in latencies with greater than 16 threads in use.

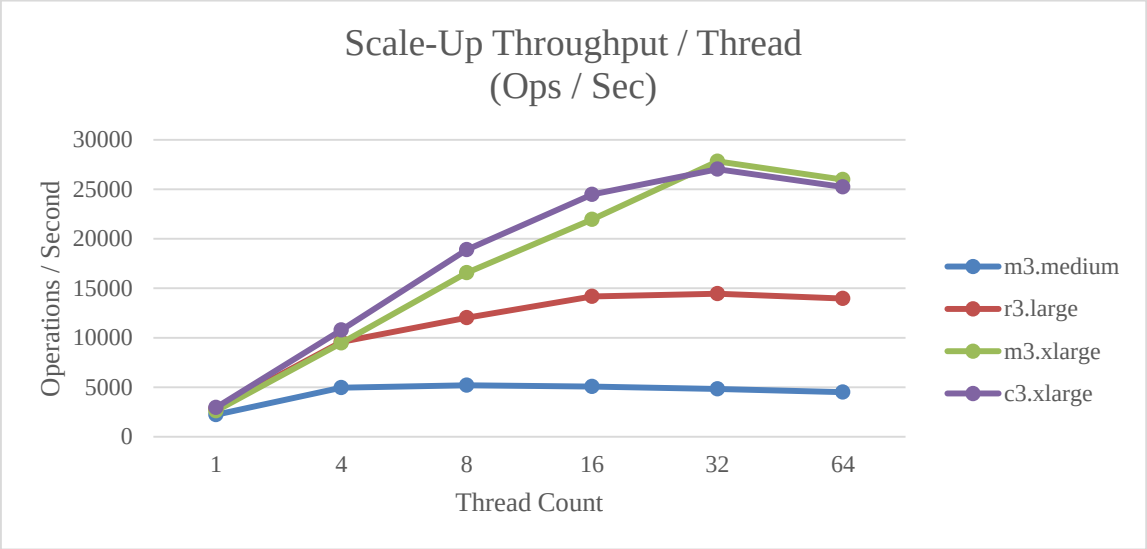

Figure 21: Scale-up Operations / Second / Thread for 1 Million operations

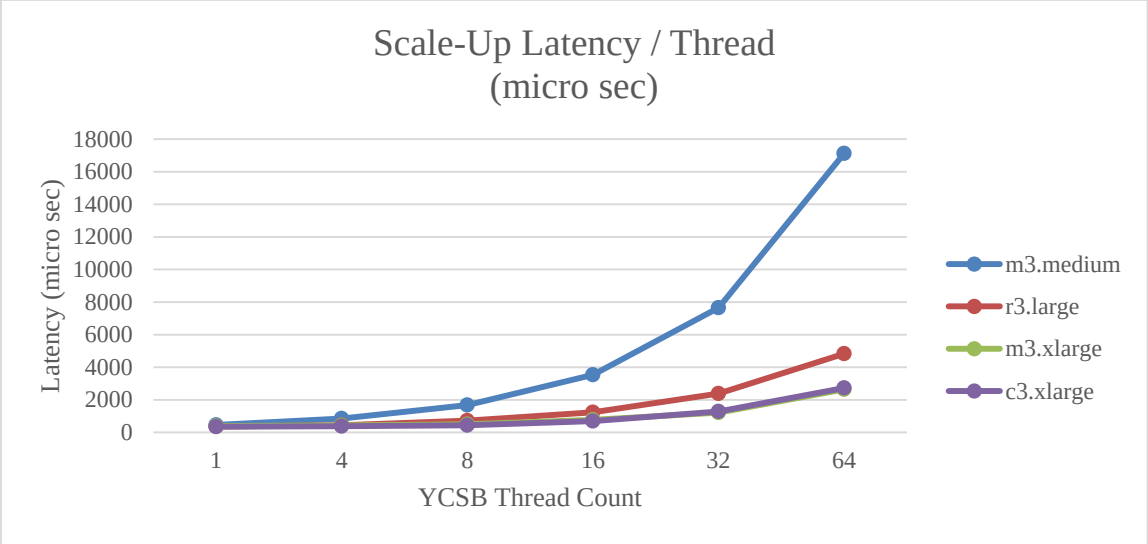

Figure 22: Scale-Up Latency / Thread for 1 Million operations (micro seconds)

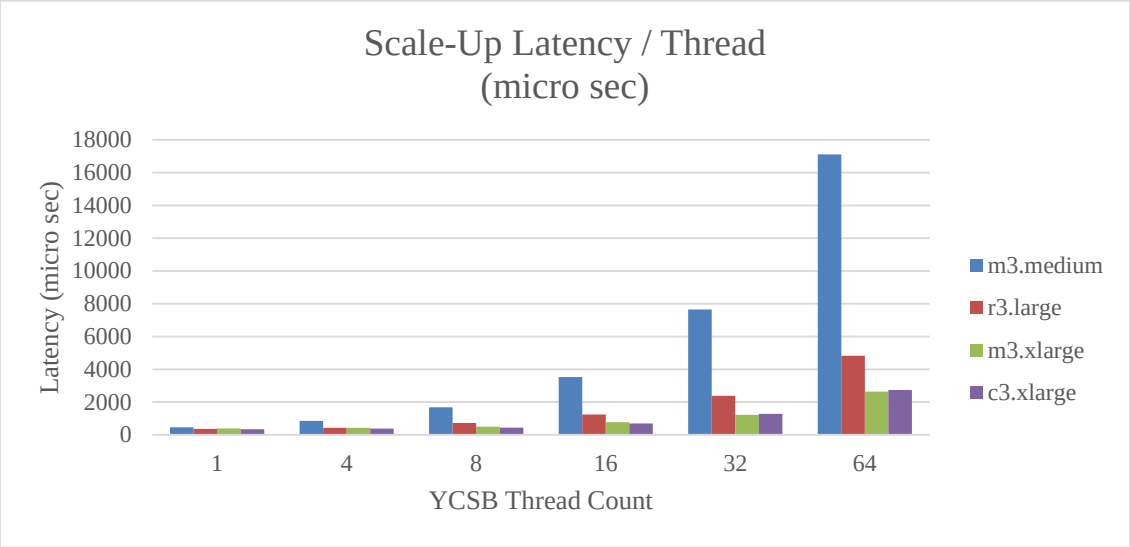

Figure 23: Scale-Up Latency / Thread for 1 Million operations (micro seconds) - Column Chart

As is shown in figure 24 below, CPU activity levels were gathered across each test instance and the YCSB benchmark instance while each test was running. The metrics were gathered using the Linux "top" command, which displays the CPU busy % as well as other metrics such as Memory % etc. When the benchmarks were being run the Memory Used % was monitored. Through this monitoring it was apparent that memory would not be a restricting limit as in all instances the memory used % never went above 25%. The purpose of gathering these metrics was to confirm and check that the limitations reached by the increasing thread count were down to the mongo test instances themselves reaching their limit of performance and not the YCSB benchmark applications instance. The % busy figures were calculated taking into account the multi-core configuration of various instances. For example, for the c3.xl instance there are 4 vCPUs, therefore the % busy reported is a sum of the activity on all 4 vCPUs. To find the overall activity level in regards to the instances capability the % CPU value is divided by the number of vCPUs (Stack Exchange, 2014). This resulting values were those that were used in the graph in figure 24 below.

The % CPU busy graph in figure 24 shows primarily that the YCSB instance never hampered the overall throughput with CPU activity levels remaining below 10% for each test. The graph does highlight that when reviewing the performance in relation to the thread count, that each instances CPU reaches its limit at some point. The performance, and latency as previously observed are directly related to the number of threads and as can be seen from the below graph, relative to the overall CPU activity level per instance. The sooner the instances CPU is close to 100% the latency is affected as the application (YCSB) must wait for CPU cycles to be freed up to perform the next workload. As the smaller instances (m3.medium and r3.large) have less available vCPUs then the activity level quickly maxes out and the latency and therefore overall throughput is affected negatively. This graph really highlights that the overall throughput, when taking into account threads, is primarily affected by the vCPU count and speed. A point to note is that this increase in threads is not relevant for every application. Applications and deployments have their own recommended thread count as

they may not have the ability to scale to a large number of threads and so would not be as positively impacted as shown here.

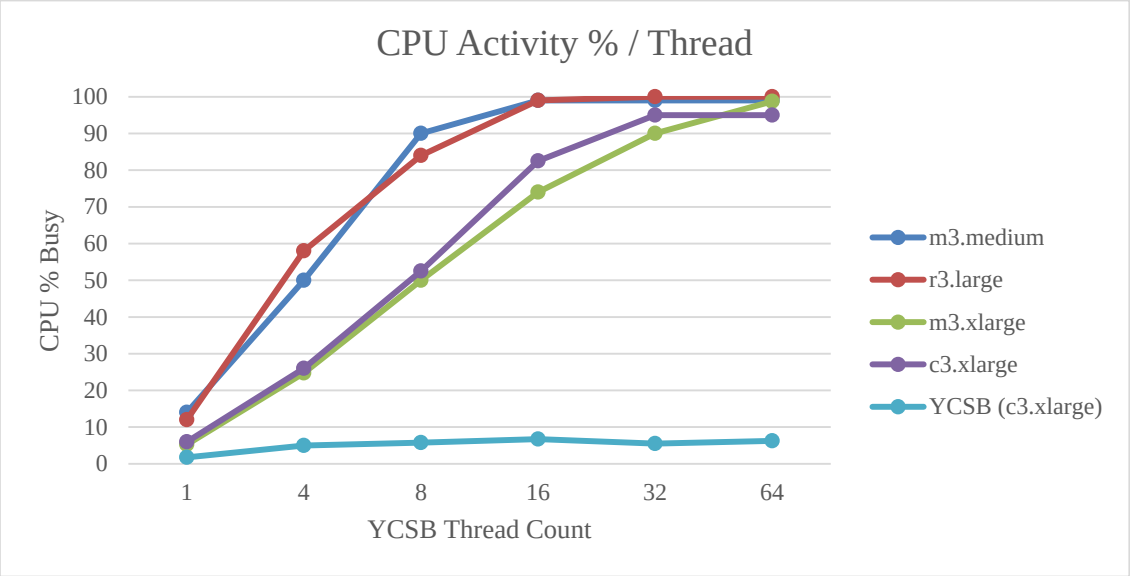

Figure 24: AWS instance CPU % Busy / YCSB Thread

Leveraging the efficiency formula (formula 5) with the multithreaded results shows a significant difference in the results for each instance as the number of threads is increased. The efficiency calculation is based on the time improvement factor with regards to the increase in ECU score. As mentioned previously, the lack of an AWS EC2 instance with 1 ECU means that the efficiency of the other instances is being measure relative to the performance of the m3.medium instance. As can be seen in figure 25 below, the m3.medium results were used as the basis for the 1 ECU reference point and so should result in an efficiency score of 1. As the other instances all had a less efficient (less than 1) score when compared with 1 thread, it can be seen that the increase in the number of threads increases the efficiency of each instance. Each of the instances become more efficient than the 1 ECU reference point at differing points on the graph. The r3.large is the second smallest from an ECU score and based on the performance improvement that it brings with the increasing thread count, it is calculated as the most efficient overall. This highlights that efficiency is not directly related to ECU count. As can be seen below, even though the instances with more compute, m3.xlarge and c3.xlarge, can perform more operations per second with increasing thread count, the increase in performance is not as significant when considering the size of the instances from an ECU score. This is a similar observation to the previous research which observed that the increase in resources available when scaling-up may not all be utilized effectively (Hwang, et al., 2016), and so, theoretically they are wasted.

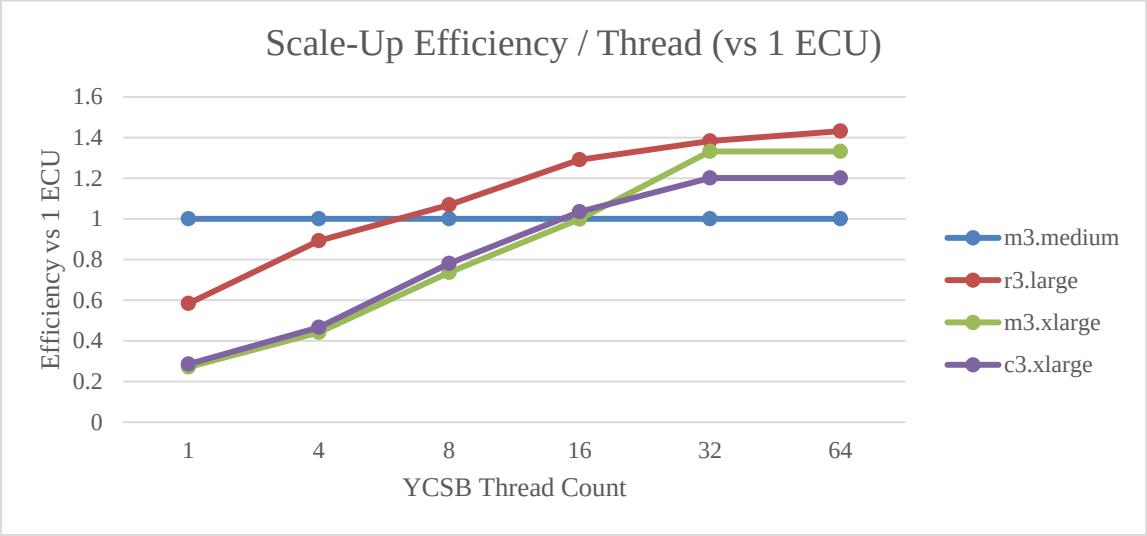

Figure 25: Scale-up Efficiency / Thread (vs 1 ECU)

The productivity of the instances similarly varies according to the number of threads being handled. The productivity formula (formula 6) is based on values gathered from the results and the cost of operation of an instance and so is an accurate guideline as to the exact cost of running each instance for the particular workload. Figure 26 below shows the productivity comparison between the various instances when threads are taken into account.

As was previously seen when benchmarking with 1 thread, the cheapest instance, m3.medium, was seen to provide the best productivity. However, when the productivity is subsequently calculated taking into account the increased performance with an increased thread count, the results look significantly different. For example, the r3.large instance was observed to be the most efficient in figure 25 above. However what can be observed is that the increased cost of the instance over the m3.medium means they have similar productivity metrics. The increased performance is in line with the increase in cost and therefore no substantial increase in productivity is gained. However the increased performance and throughput of the more compute centric instances of m3.xlarge and c3.xlarge means that they become more productive as the threads increase. This can be understood to be due to their ability to handle the increasing workload coming from the multi-threaded application better. The c3.xlarge instance confirms that when considering which instance to deploy a workload onto there is many different aspects to consider simultaneously. In the productivity metrics the largest c3.xlarge instance is by far the best considering that all of the other information gathered (throughput, latency, and efficiency) point to 16 threads being close to the sweet spot for this workload.

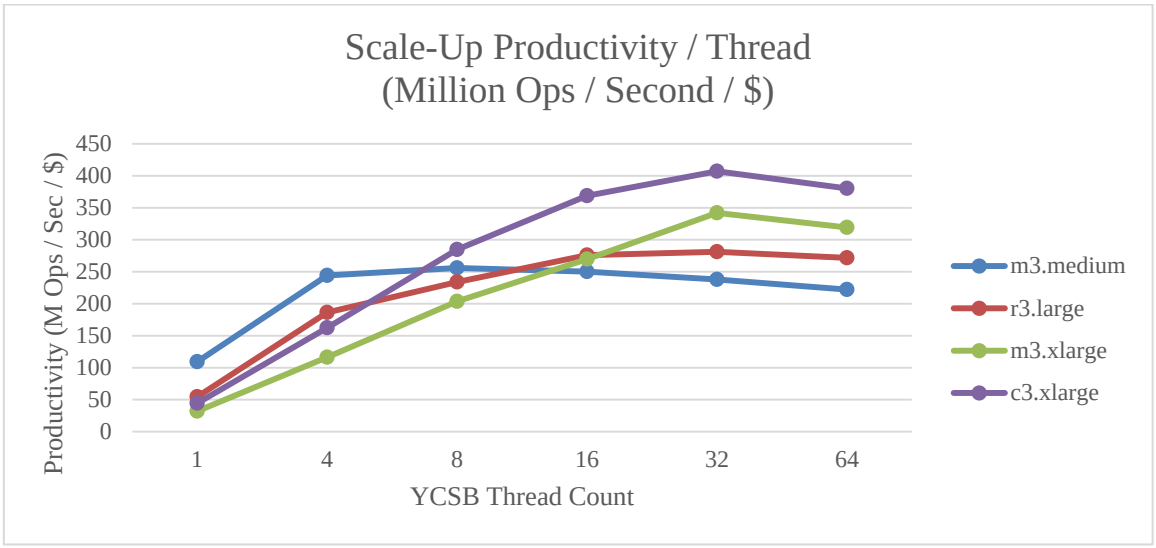

Figure 26: Scale-up Productivity / Thread (M Ops/Sec/$)

4.2 Scale-Out overview

4.2.1 Configuration

To create a scale-out environment a number of fundamental changes to the deployment was required. To enable a scale-out MongoDB cluster to be created the steps outlined in the MongoDB documentation needed to be followed appropriately (MongoDB, 2017). An image representing the general configuration is shown in figure 27 below.

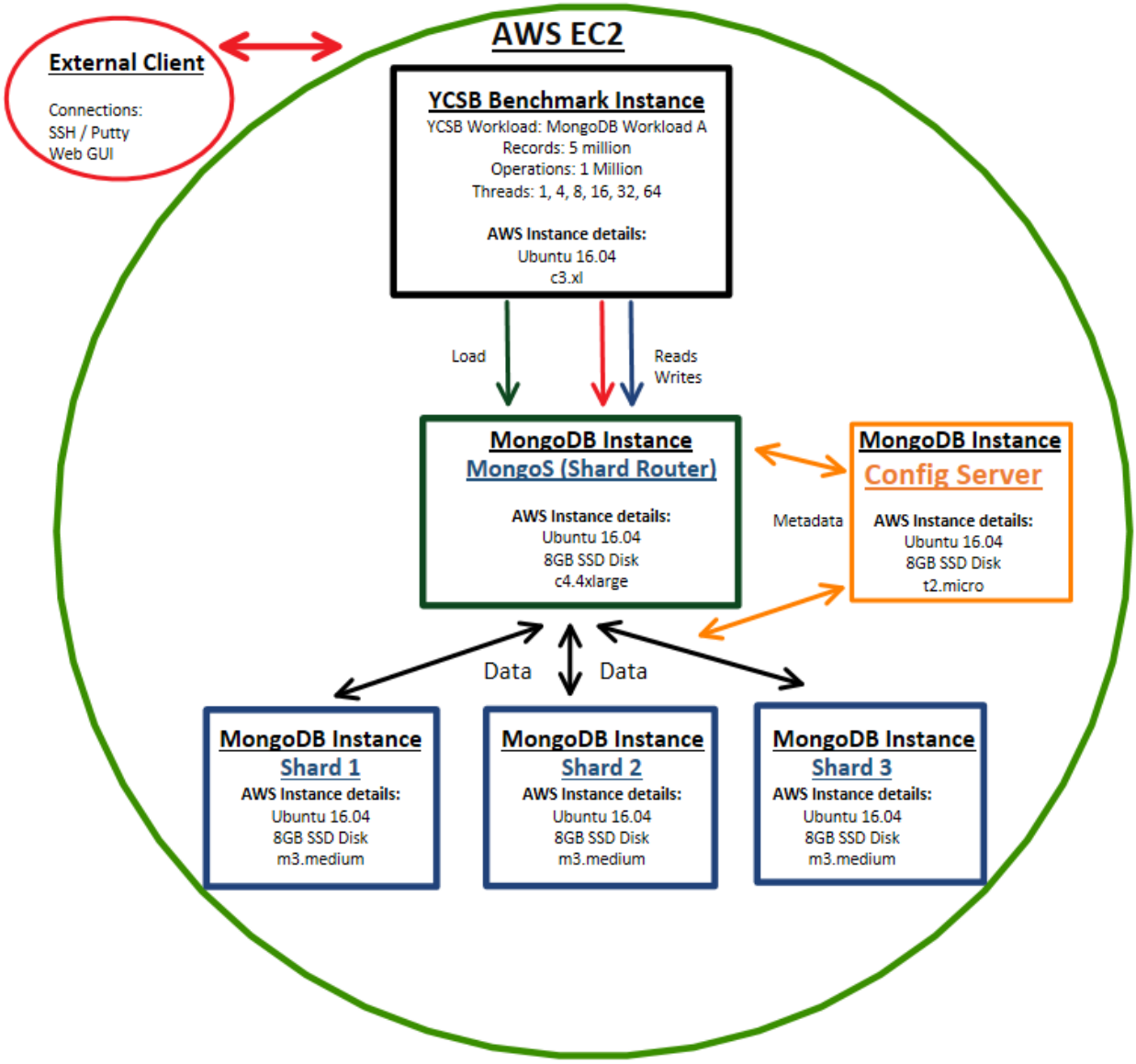

Figure 27: Scale-out instance configuration

Similarly to the scale-up deployment outlined in the scale-up configuration section above, there was no creation of replica sets for each mongo instance for the scale-out configuration. The instance size m3.medium, was used for all mongo servers contributing resources as shard instances within the cluster. This instance was chosen as it was the smallest AWS instance available with predictable performance and ECU metrics. As it

was the smallest instance with the least performance and memory capabilities it enabled the benchmark tests to push it to its performance limit. This then allowed the results to accurately record the benefit that increasing the number of instances would have by each reaching their performance limit at different stages. If a more performant instance was used in place of the m3.medium, the YCSB benchmark and MongoDB Shard router instances may not have been able to impart a workload high enough to push the sharded cluster to its limit. The choice of m3.medium also enabled a direct comparison to be made to both the results for m3.medium gathered in the scale-up results and also the previous research reviewed (Hwang, et al., 2016).

The MongoDB shard router (mongos) was initially built on this r3.large instance also but the early results indicated that as the number of YCSB threads increased that this quickly became a bottleneck in the configuration. This instance was therefore increased to a much larger compute intensive instance, c4.4xl. This change ensured that the shard router was not impacting the tests being done against the scale-out shard instances. As the MongoDB Config Server was only storing the configuration and not in the data path with any ability to affect the overall results, it was built on a t2.micro server to keep costs down. A single MongoDB Shard router (mongos) instance was deployed to reduce complexity and cost. This single instance was deemed sufficient as the number of shards was not being expanded large enough to require a second for load balancing.

When scaling out, the number of shard servers was increased from 1, to 2 to 3 respectively. To ensure that there was a more granular spread of data "shards" across the small number of shard instances, the shard chunk size was configured as 8Mb (MongoDB, 2017). This was reduced from the default chunk size of 64Mb to ensure that there was an 8 times increase in the number of shards as initial indications suggested there would only be ~10 shards created. The remainder of the environments configuration remaining the same.

The YCSB deployment was created in the same fashion and with the same configurations as outlined in the scale-up deployment section 4.1.1 above. No changes were made to enable a similar benchmark test be performed across both scale-up and scale-out.

4.2.2 Results observed

The calculations were done using the metrics gathered from the 1 million operations, multi-threaded benchmark runs. The results of one set of operation values was done to provide an easier to interpret graph. The similarity between the initial 1 threaded 1, 5 and 10 million operation runs meant that there would not be any obvious impact from choosing one operation count over the other.

4.2.2.1 Scale-out: Observations on increasing YCSB thread count

Figure 28 shows the throughput of each MongoDB sharded cluster as it scales out from 1 shard to 3 shards. The predictable increasing operations per second is what would be expected to be seen with scale out

configurations and as previously mentioned, scalability and performance are two of the key reasons for adoption of cloud technology. All three configurations use the same building block of the m3.medium instance and appear to peak at between 8 and 16 threads. This is a similar point to that observed when the m3.medium instance was tested during the scale-up deployments. It may have been expected to see the clusters with 2 and 3 shards peaking with more threads as the clusters have additional CPU resources. As the m3.medium instance only consists of 1 vCPU then the overall increase in compute power may not provide sufficient processing ability to handle a thread count above 8 overall. Furthermore, as the deployment means that there is additional network communication required between the instances, the CPU scale-out benefit may have been impacted by this additional latency. Additional review of the capabilities of CPU clustering and the impact of networking is outside the scope of this paper but will be noted for future investigation.

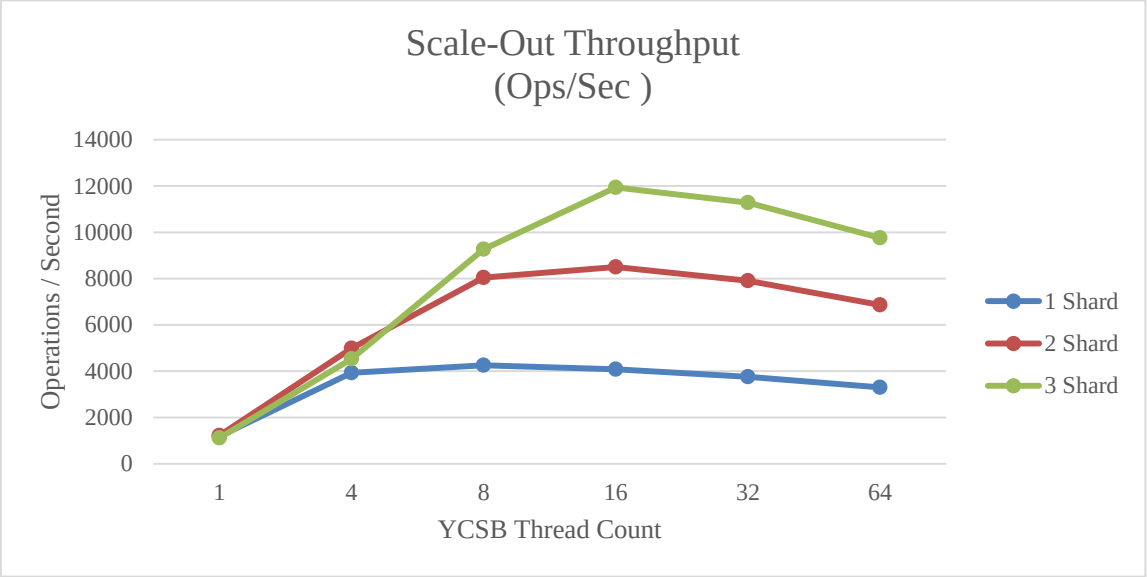

Figure 28: Scale-out Throughput / Cluster / Thread

As mentioned in the scale-up configuration, the overall throughput shown in figure 28 above, needs to be viewed in conjunction with the relative update latency, shown in figure 29 below. The throughput and latency are inversely proportional as the latency impacts the overall throughput negatively as the applications wait time between operations increases. As can be observed from the latency graph below, a significant increase begins to occur when the thread count rises above 8 threads.

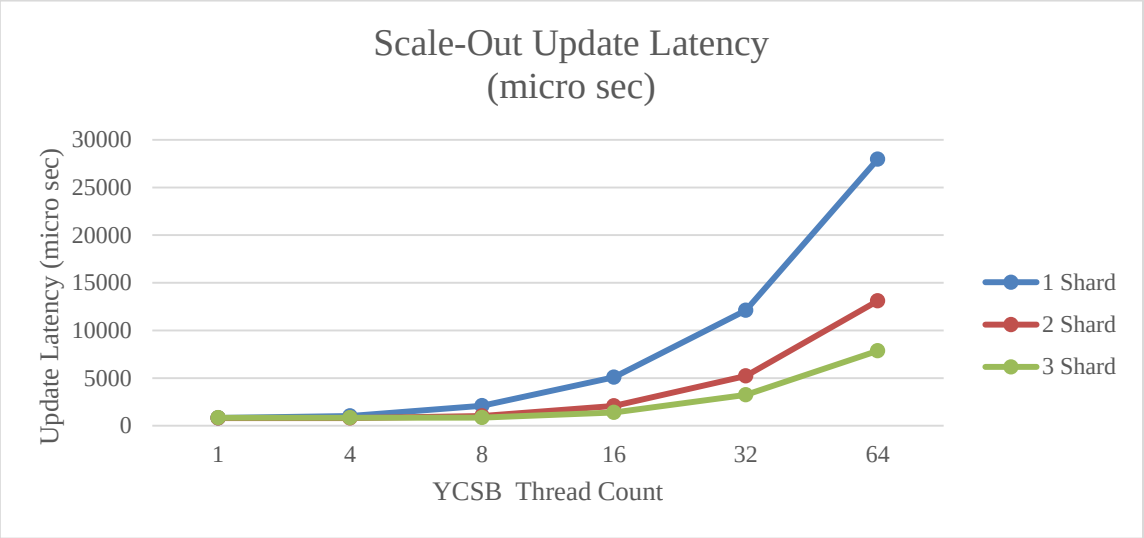

Figure 29: Scale-out Update Latency / Thread

In the same way as reviewed in the scale-up results, the activity levels of the CPU and memory of each instance were measured as the threads increased to determine if there was a particular configuration aspect which was impeding performance. As the configuration of the scale-out system involved many additional instances including the mongos (router) and the configuration server as well as the YCSB and shard cluster, the metrics of each was gathered for each different thread value tested. The used memory levels across all instances were seen to be below 30% and so have not been graphed as there would be no tangible impact to the performance. The primary objective as with all benchmarking was to once again ensure that any performance bottleneck was occurring at the relevant system under test. In the scale-out scenario, the MongoDB shard level was the system under test where it was necessary to see the bottleneck, if anywhere. As can be seen from figure 30 below, the CPU activity level is maxed out at 16 threads for 1 shard instance. This would impact the performance and can be seen as a point in which there is a significant increase in the latency graph above. As observed in figure 28 above, the performance continues to increase to the peak of 16 threads when the number of shards increases to 2 and 3 but then reduces. As observed in the CPU activity graph in figure 30 below, this performance reduction and latency increase does not coincide with the instances maxing out. As the load is spread across increasing number of instances the CPU did not reach 100% while being observed. This reduction in the performance and increase in the latency points to a limit being reached. While the instance still has capacity for additional CPU cycles the results would point to another cause of performance reduction. As mentioned previously, this could potentially be due to the increasing impact of the network over the configuration or some other instance. This would warrant further investigation into the limits of AWS / MongoDB sharded clusters which is outside the scope of this project.

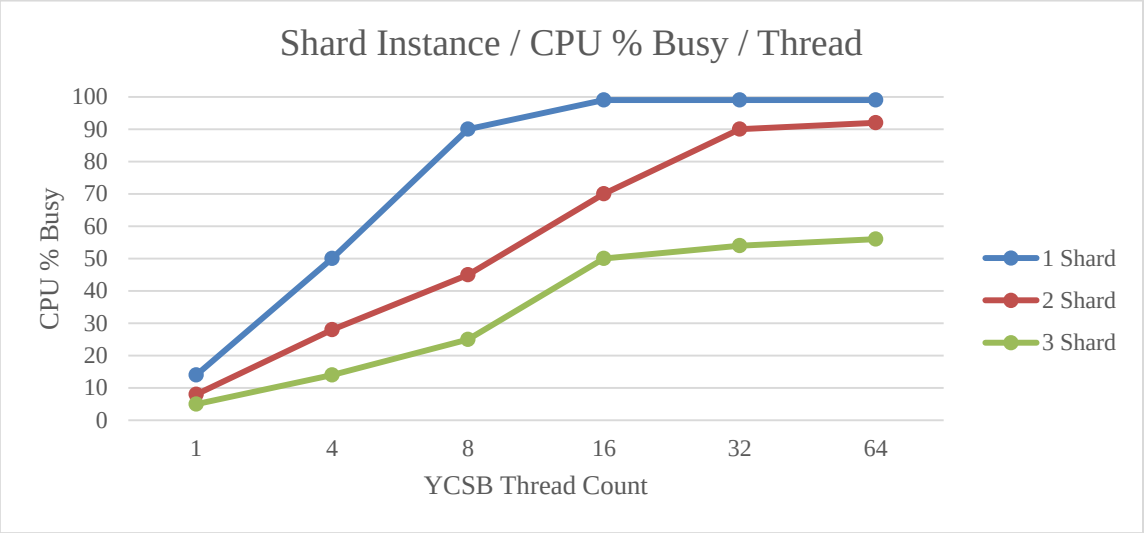

Figure 30: CPU activity level per Shard Instance

The mongos (shard router), YCSB and MongoDB Configuration Servers metrics were also gathered. As detailed, to ensure that there was no observed bottleneck which could be mitigated with a larger AWS instance. Figures 31, 32 and 33 show the CPU activity levels for each increasing shard configuration. Each figure clearly shows that the CPU levels of these instances were not of concern when considering potential bottlenecks to the overall performance as no instance reached a CPU activity level above 50%.

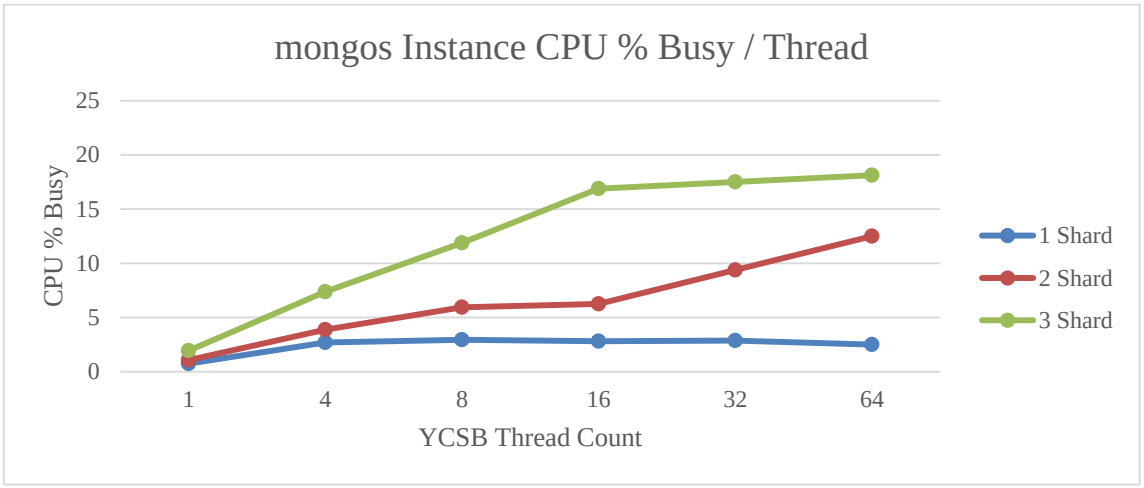

Figure 31: mongos Instance CPU activity level

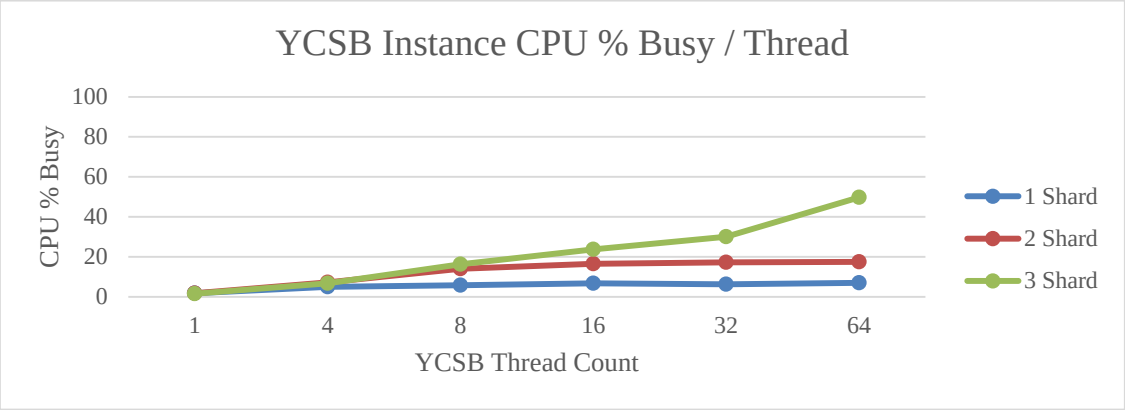

Figure 32: YCSB Instance CPU activity level

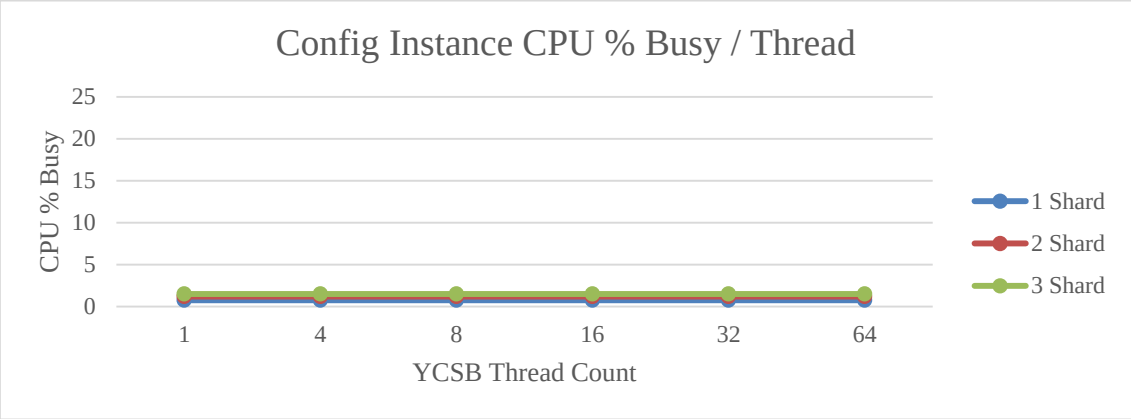

Figure 33: MongoDB Config Instance CPU activity level

The efficiency of the scale-out configurations is measured using the same formula (formula 5) as previously used in the scale-up analysis. The same 1 ECU time calculated from running the operations against the m3.medium instance in the scale-up benchmarking was used as a basis for comparison. This value was used to ensure that the sharded configuration, with the additional requirements and data paths through the mongo shard router, config server etc. could be compared against a standalone non-sharded 1 instance configuration. As can be seen in scale-out efficiency graph in figure 34 below, the starting efficiency of the configurations is lower when the thread count is lower. It is not until the thread count increases to that observed as the peak in the throughput / performance graph in figure 28 above, that the efficiency also reaches its peak and is similar across the three configurations. At no point does the efficiency increase to more than 1 in the scale-out configuration whereas it does for all instances in the scale-up testing. As the efficiency is based on the speedup of the operations relative to the overall ECU count, the larger the number of instances in the scale-out configuration, the greater the resources configured and the more potential there is for wasted resources / reduced efficiency. A similar observation was made in the previous research, where from both a scale-up and scale-out perspective there was a decrease in efficiency observed (Hwang, et al., 2016).

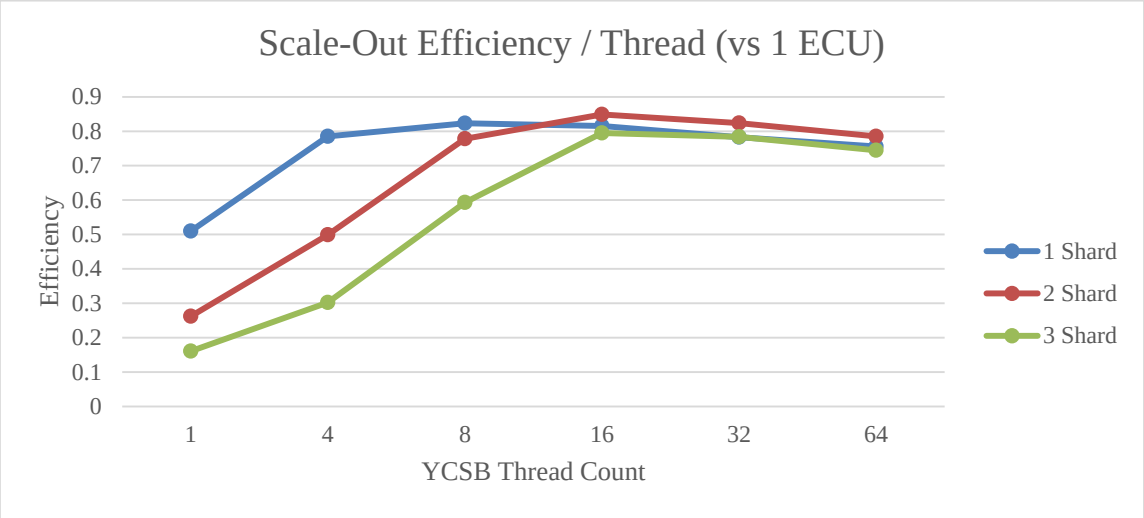

Figure 34: Scale-out efficiency / thread vs 1 ECU

The final graph shown in figure 35 below, displays the productivity of each scale-out sharded configuration. This calculation is based on formula 6 and uses the same assumptions regarding the system availability of AWS as detailed in the productivity formula definition.

In a similar trend to that observed in the efficiency graph above, all configurations reach peak productivity at their corresponding throughput peak as expected. The increasing cost of the additional instances used in the scale-out configurations impacts the productivity until the throughput reaches its peak. As was observed in the throughput results earlier, the addition of an instance to the scale-out configuration brings with it an additional increase in performance. The fact that each configuration delivers a similar peak productivity indicates that should the configuration be scaled-out further that the productivity would remain the same. This would be down to the increase in overall throughput at a similar trend to the increase in the cost of deployment. As was observed in the latency results also, the further the deployment scales out, the lower the latency becomes. This productivity metric highlights that the more money that is spent on scale-out resources, the more throughput will be generated and the lower the latency observed. This provides a predictable method of meeting requirements from both a performance and cost perspective.

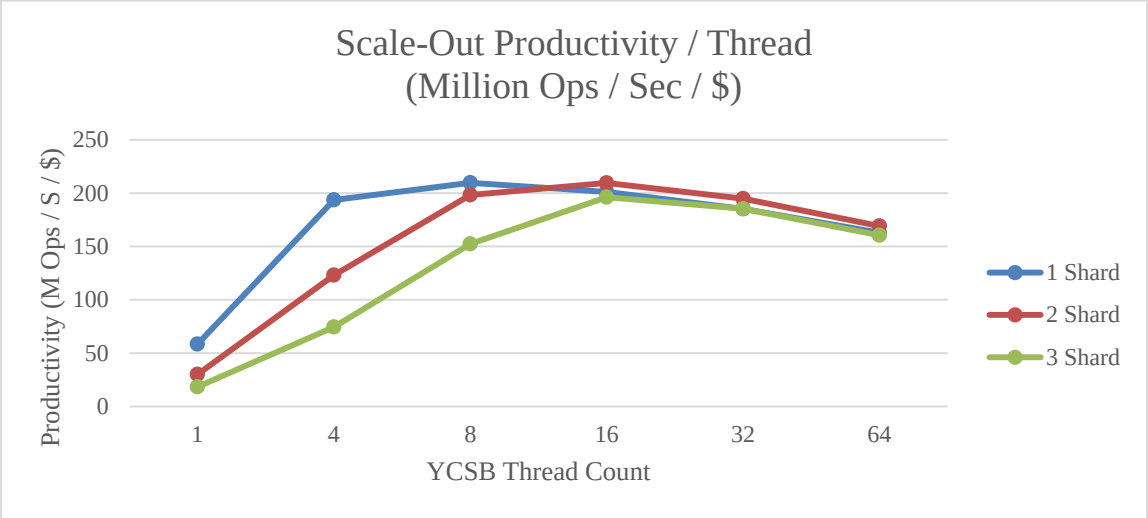

Figure 35: Scale-out productivity per thread (million operations / second / $)

5 Conclusion

5.1 Observations

As mentioned at the outset, the purpose of this research was to investigate and review the difference in impact that some of the inhibitors to cloud performance could have relative to the available scaling strategies. These objectives were created following the review of factors influencing the adoption of cloud technologies. The particular factors that were chosen for further research, due to their impact on cloud consumer sentiment, were performance and cost efficiency. The review of these particular adoption influencers directed further research into the means and ease by which cloud consumers could gather facts and make informed decisions on cloud adoption for their particular enterprise. The benchmarking completed and metrics used was influenced by the desire to build on the recent and relevant research carried out (Hwang, et al., 2016). Further research into whether there are particular benefits to be observed from deploying either a scale-up or scale-out configuration was also investigated to build on other research (Appuswamy, et al., 2013).

This research has used metrics, which were developed in previous papers (Hwang, et al., 2016) (Hwang, et al., 2014), in order to understand, measure and report on the performance and efficiency of scale-up and scale-out configurations. Following the collection of the benchmark results and the calculations of the specific metrics it was possible to perform an overall comparison between scale-up and scale-out. The results showed that following the tests produced in this research, scaling-up was a more efficient / productive means of running the MongoDB workload. This can be observed in the overall performance, efficiency and the productivity graphs where the resulting figures of the "best" instance overall (c3.xlarge) are each better than those observed for the "best" configuration during the scale-out testing. The "best" is that which was chosen to represent the configuration with a balance of throughput, latency, efficiency and productivity but may not be "best" for all application deployments. For example, for the scale up instance (c3.xlarge) there was a maximum efficiency value of 1.2 observed and maximum productivity of 407 million operations / sec / $ in comparison to the best scale out deployment (2 shard) where a maximum efficiency value of 0.84 and productivity of 209 million operations / sec / $ was reached. The understanding is that this difference is seen due to the ability to choose a particular scale-up instance which will be advantageous to the exact workloads requirements. In this case the scale-up instance configured (c3.xlarge) was sufficiently compute and memory capable to provide good overall throughput and response time to the YCSB Workload A benchmark run. Depending on the specific applications requirements; the number of vCPUs required / threads, the workload size, the latency limit, etc. there is the potential to scale-up to a single instance with close to the exact recommended specifications. This is a similar observation as was made in previous research on scale-up vs scale-out for Hadoop (Appuswamy, et al., 2013). This means that when scale-up instances are chosen and deployed accurately there is limited over configuration and as previously observed, a highly efficient deployment. Should the workload or application requirements then change there is the potential that the deployment is under or over configured and either the applications performance or the efficiency /

productivity figures will be negatively impacted. This is a key negative with the use of scaling-up, the lack of ease of scaling from one instance to another. This additional difficulty means that there is often instances which are under (such as m3.medium) or overprovisioned with resources which affects either the performance or cost / efficiency negatively.

Scaling-up has been observed to provide a more adaptable and adjustable configuration which could be scaled-up or scaled-down relative to the particular requirements at the time. There is more time required to deliberate on the different and best options when deploying a scale-out configuration as there are typically more parts and variables to consider. However, as each instance could be seen as a building block of sorts then there is the capability to much more easily and granularly adjust. Increasing the number of instances was seen to negatively impact the overall efficiency of the scale-out configuration in this research. The overall throughput increased from 4252, 8497 to 11939 operations/second through the increasing cluster size of 1, 2 and 3 shards. Simultaneously the latency decreased consistently with the increase in scale-out cluster size. It dropped from a peak of 27,000 through 13,100 down to 7,900 micro seconds for 64 threads across the 1, 2 and 3 shard configurations. So this reduction in efficiency, while technically wasting provisioned resources, provided for much better performance. As the productivity of the increasing scale-out cluster was consistent when scaling, the overall observation was that the increasing cluster size would have a positive impact on the cost to serve particular applications through the added benefit of a reduction in latency.

This research built on previous work by considering the impact of increasing threads on the benchmark application. The overall expectation that an increase in the thread count would increase the performance matches the observed outcome. In the configurations and benchmarking performed in this research it was observed that there was a similar peak in throughput and spike in latency when the number of threads reached 16. As mentioned at the outset, this particular thread count is only relative for this configuration and use case as different applications and configurations would be expected to have different recommendations. The research previously carried out only performed benchmarking with 1 thread and as has been observed in this paper that is consistently the worst performing configuration. In this research the impact of increasing the threads was measured to highlight the fact that the actual productivity and efficiency of cloud deployments depends on many factors. The overall applications performance and efficiency needs to be able to take into account the impact of all possible configuration options of cloud instances and deployments, including the usage of multiple threads. This is similar to that discussed in the cloud adoption factors earlier where consumers do not actually consider all of the metrics and costs which impact the overall cost of adoption. Incorrectly calculating these figures may results in a cloud strategy being adopted or dismissed incorrectly and being deemed a success or failure inaccurately.

This research additionally attempted to use the same configuration / deployment for both scale-up and scale-out. Previous research had used different applications, benchmarking tools, number of instances, metrics etc. and attempted to make an educated guess in order to gather overall trends across these different deployments.

As highlighted previously the ability to use one application (MongoDB) and one benchmarking tool (YCSB) across both scale-up and scale-out enables for an accurate comparison. As detailed in the configuration steps, it was not possible to make the configurations identical and so the results cannot be directly compared. The fact that there is a slight difference does not mean that the results cannot be accurately compared however as they are based on the same underlying configuration and deployment. Viewing the difference between the performance of the m3.medium instance from both a sharded and non-sharded deployment provides the ability to see the additional overhead that sharding adds to the configuration.

Reviewing the throughput, efficiency and productivity graphs highlight the similarities and differences between each deployment option from a measured view. From first glance it would appear that there is better efficiency and productivity, not to mind a less complex deployment, when MongoDB is on a single large instance. What is not considered in these metrics is the additional benefits / restrictions to each type of deployment. Scaling-up does not give all of the same advantages that scaling-out does. In the case of MongoDB for example, running on a single instance and scaling-up means that many of the benefits of cloud, such as fault tolerance and scalability are lost. Having sufficient memory and performance configured in the deployment is key to having a well running application but the benefit of scaling-out is that if the deployment needs to be changed there is an unlimited pool of resources to scale-out into typically.

Advantages of scale-up over scale-out:

- Easier to implement as there is only a single instance to configure and manage.
- More cost effective and productive due to being able to choose specific option.
- Hard to scale, there could be a negative impact to the application if changing the size of the instance.

Advantages of scale-out over scale-up:

- Easier to scale as additional instances can be deployed and the cluster can increase / decrease if required.
- Can be implemented with auto-scaling capabilities removing some management responsibility.
- Granular and predictable performance as additional scaling brings specific resources.
- Encompasses more of the benefits of cloud; scalability, flexibility, high availability and business continuity.

Overall the findings of this research broadly reinforce those in the previous research carried out (Hwang, et al., 2016). As stated previously in both this and other research, the benchmarking completed provides metrics for the system under test in those configurations but real-world deployments may be different and provide different results. The particular trends observed for performance and efficiency were similar to those observed in other research. As was seen earlier, there are scale-up instances which offer better overall performance and

efficiency metrics in comparison to scale-out deployments. Scale-out offers a predictable and deliverable increase in performance and productivity.

5.2 Further Considerations and Research

Some areas of the research presented in this work would require further investigation and review in order to provide a more comprehensive conclusion and ensure that all concerns and issues were addressed. The scope of this project meant that there are some items which have been identified as areas for further consideration and research.

This paper built on research previously completed around cloud scaling options and auto-scaling techniques (Hwang, et al., 2014) (Hwang, et al., 2016). Auto-scaling was not used due to concerns raised in these papers around the "elasticity" (a function of the reconfiguration overhead) of the cloud. Auto-scaling is one of the key benefits of using scale-out technologies. Auto-scaling enables the concept of provisioning what is required in order to reduce the size and cost of deployment and increase the efficiency and productivity. Scaling-up is deemed as having a high overhead for reconfiguration but the overhead for scale-out also needs to be accounted for. As technology changes there may be increased ability and speed enabling scale-out, scale-up, or mixed scaling to be done more efficiently and effectively. This means that a combination of technologies may be easier to configure and manage. Further research into the cost implications that the down-time or management involved when scaling-up and scaling-out should be completed.

For this reason the efficiency of auto-scaling technologies should be reviewed. There may be less of a brute-force impact seen when scaling as was previously observed (Hwang, et al., 2014). This research may then produce a more effective means of delivering performance as the algorithm and overheads required improve.

As observed in both the scale-up and scale-out benchmark results, there is negligible difference between the throughput observed for the increasing operations counts of 1, 5 and 10 million. This is in contrast to the previous research where, for example, benchmarking HBASE produced significant differences in the throughput for the increasing workloads (Hwang, et al., 2016). The differences between the various benchmark results provides scope for further research to be carried out. Investigations into the requirements and benefits of increasing the operations for the different applications, with MongoDB being of particular interest due to it producing unanticipated results.

A review of the reason that the m3.xlarge instance performed so poorly in the initial scale-up, 1 thread benchmarking would also provide some further answers. What was unexpected was the significant difference between the m3.xlarge results and those observed for the r3.large and c3.xlarge instances. The m3.xlarge provided approx. 2500 ops/sec in comparison to the approx. 3000 and 2850 ops/sec observed for the r3.large

and c3.xlarge instances respectively. As highlighted previously, the m3.xlarge instance has a similar memory configuration and ECU score as the r3 and c3 instances respectively so it is not apparent why the throughput varied so significantly. When further YCSB threads were implemented the m3.xlarge instance recovered to perform as well as if not better than the other instances. Additional research into the results observed with 1 thread and why there was a difference would provide significant benefit.

When considering the costs and metrics for efficiency and productivity in this paper, there were a number of items which were not taken into account. Primarily the cost of the additional instances required to deploy and run the scale-out configuration of MongoDB was not taken into account. As mentioned in the scale-out section, the reason for this was an effort to compare the performance and cost of hosting the actual workload and comparing the metrics of running it on 1 instance size compared to another, further compared to running it on multiple instances. The additional costs of running the other instances was not included. Further research should be done comparing the cost of the overall deployments to give a clearer picture overall.

Comparing the scale-up and scale-out configurations results as discussed previously, one other aspect would benefit from further research to build on the observations. The throughput showed that when deploying the instances in either scale-up or scale-out configurations there appeared to be significant loss of throughput for the scale-out deployments. As the same instance, m3.medium, was used as the basis for both configurations this could be seen more clearly. The 1 instance (scale-up) deployment managed a peak of approx. 5200 ops/sec, whereas the 1 instance (scale-out) sharded deployment reached a peak of approx. 4250 ops/sec. This is approx. 18% reduction in performance with the same AWS instance type being used. This difference would have significantly impacted the subsequent productivity and efficiency metrics and the overall conclusion. The cause for the loss of throughput would appear to be down to the additional overhead created by the extra instances and processes required to maintain a sharded MongoDB configuration. Further investigation into what overhead, if any, each of the shard router, configuration server and sharding process add would build on this research and may enable additional conclusions be made.

Finally, due to the scope and the knowledge of the researcher, this thesis provided comparative research between a relatively small set of AWS instances and a small scale-out configuration. The purpose was to show the differences and compare the metrics of both performance and cost efficiency. To build on this research additional scale-out and scale-up configurations should be investigated to discover if larger deployments have an impact on the results.

6 Bibliography

Accenture, 2015. *Enterprise Cloud Report 2015,* s.l.: s.n.

Accenture, 2016. *Perception or Reality? The Truth About Cloud,* s.l.: s.n.

Amazon Web Services, 2016. *AWS Security Whitepaper.* [Online]
Available at: https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf
[Accessed 10 10 2016].

Amazon, 2016. *Amazon Virtual Private Cloud (VPC).* [Online]
Available at: https://aws.amazon.com/vpc/
[Accessed 09 10 2016].

Andrea Zanella, N. B. A. C. L. V. M. Z., 2014. Internet of things for smart cities.. *IEEE Internet of Things journal,* 1(1), pp. 22-32.

Armbrust, M. et al., 2010. A view of cloud computing. *Communications of the ACM,* 4(53), pp. 50-58.

Avram, M.-G., 2014. Advantages and challenges of cloud computing from an enterprise perspective. *Procedia Technology,* Volume 12, pp. 529-534.

Benlian, A. & Hess, T., 2011. Opportunities and risks of software-as-a-service: Findings from a survey of IT executives.. *Decision Support Systems,* 52(1), pp. 232-246.

Beserra, P. V. et al., 2012. *Cloudstep: A Step-by-Step Decision Process to.* Toronto, IEEE, pp. 7-16.

Bhardwaj, S., Jain, L. & Jain, S., 2010. Cloud computing: A study of infrastructure as a service (IAAS). *International Journal of engineering and information Technology,* Issue 2.1, pp. 60-63.

Boonchieng, E., 2014. Performance and security issue on open source private cloud.. *Electrical Engineering Congress (iEECON), 2014 International,* 19 03, pp. 1-5.

Carroll, M., Merwe, A. V. D. & Kotze, P., 2011. *Secure cloud computing: Benefits, risks and controls.* s.l., IEEE.

Cloud Security Alliance, 2016. *The Treacherous 12 - Cloud Computing Top Threats in 2016.* [Online]
Available at: https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf
[Accessed 26 01 2017].

Daniele Miorandi, S. S. F. D. P. I. C., 2012. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks,* 10(7), pp. 1497-1516.

David G. Andersen, J. F. M. K. A. P. L. T. V. V., 2009. *FAWN: A fast array of wimpy nodes.*. s.l., ACM, pp. 1-14.

ENISA, 2009. *Cloud Computing - Benefits, risks and recommendations for information security.* [Online]
Available at: https://www.enisa.europa.eu/publications/cloud-computing-risk-assessment
[Accessed 28 01 2017].

ENISA, 2016. *Cloud Computing Risk Assessment.* [Online]
Available at: https://www.enisa.europa.eu/publications/cloud-computing-risk-assessment
[Accessed 09 10 2016].

Enno Folkerts, A. A. K. S. A. I. V. M. C. T., 2012. *Benchmarking in the Cloud: What It Should, Can, and Cannot Be.*. s.l., Springer Berlin Heidelberg, pp. 173-188.

EU, 2016. *EU Data Protection Reform.* [Online]
Available at: http://ec.europa.eu/justice/data-protection/reform/index_en.htm
[Accessed 10 10 2016].

Everest Group, 2014. *Enterprise Cloud Adoption Survey 2014.* [Online]
Available at: http://www.everestgrp.com/wp-content/uploads/2014/03/2014-Enterprise-Cloud-Adoption-Survey.pdf
[Accessed 1 10 2016].

Forbes, 2016. *Roundup Of Internet Of Things Forecasts And Market Estimates, 2016.* [Online]
Available at: http://www.forbes.com/sites/louiscolumbus/2016/11/27/roundup-of-internet-of-things-forecasts-and-market-estimates-2016/#65bc00e64ba5
[Accessed 11 02 2017].

Gartner, 2014. *Gartner Highlights the Top 10 Cloud Myths.* [Online]
Available at: http://www.gartner.com/newsroom/id/2889217
[Accessed 11 02 2017].

Gartner, 2014. *Gartner IT Glossary.* [Online]
Available at: http://www.gartner.com/it-glossary/bimodal/
[Accessed 7 May 2016].

Gartner, 2015. *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015.*. [Online]
Available at: http://www.gartner.com/newsroom/id/3165317
[Accessed 11 02 2017].

Gartner, 2015. *Hype Cycle for Cloud Computing, 2015*, s.l.: Gartner.

Gartner, 2016. *15 Reasons not to migrate your Data Center to Public Cloud IaaS,* s.l.: Gartner.

Gartner, 2016. *Can you save money migrating to Cloud IaaS,* s.l.: Gartner.

Gartner, 2016. *Gartner Says Worldwide Public Cloud Services Market to Grow 17 Percent in 2016.* [Online]
Available at: http://www.gartner.com/newsroom/id/3443517
[Accessed 11 02 2017].

Gartner, 2016. *Magic Quadrant for WAN Optimization.* [Online]
Available at: https://www.gartner.com/doc/reprints?id=1-36UZLWA&ct=160517&st=sb
[Accessed 22 10 2016].

Haislip, A., 2012. *Breaking Through Cloud Addiction - TechCrunch.com.* [Online]
Available at: https://techcrunch.com/2012/12/01/netflixs-amazon-cloud-addiction/
[Accessed 28 01 2017].

IDC, 2016. *Enterprise Adoption Driving Strong Growth of Public Cloud Infrastructure as a Service.* [Online]
Available at: https://www.idc.com/getdoc.jsp?containerId=prUS41599716
[Accessed 11 02 2017].

Infor, 2015. *Security of Cloud vs On Premise Deployments.* [Online]
Available at: http://www.infor.com/content/industry-insights/security-of-cloud-vs-on-permise-deployments.pdf/
[Accessed 11 02 2017].

Jackson, K. R. et al., 2010. *Performance analysis of high performance computing applications on the amazon web services cloud.*. California, Cloud Computing Technology and Science (CloudCom), pp. 159-168.

Jamshidi, P., Ahmad, A. & Pahl, C., 2013. Cloud Migration Research: A Systematic Review. *IEEE TRANSACTIONS ON CLOUD COMPUTING,* 08 October, 1(2), pp. 142 - 157.

Jerry Gao, X. B. W.-T. T., 2011. Cloud testing-issues, challenges, needs and practice.. *Software Engineering: An International Journal,* 1(1), pp. 9-23.

Kai Hwang, X. B. Y. S. M. L. W.-G. C. Y. W., 2016. Cloud performance modeling with benchmark evaluation of elastic scaling strategies.. *IEEE Transactions on Parallel and Distributed Systems,* 27(1), pp. 130-143.

Kai Hwang, Y. S. X. B., 2014. *Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity.*. s.l., IEEE, pp. 763-768.

Kavis, M., 2014. *Forbes.* [Online]
Available at: http://www.forbes.com/sites/mikekavis/2014/09/15/top-8-reasons-why-enterprises-are-passing-

on-paas/#5822e72730b1
[Accessed 09 10 2016].

KPMG, 2012. *Exploring the Cloud - A global study of Governents' adoption of Cloud.* [Online]
Available at:
https://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/exploring-cloud.pdf
[Accessed 09 10 2016].

KPMG, 2013. *The cloud takes shape - Global cloud survey: the implementation challenge,* s.l.: KPMG.

KPMG, 2014. *KPMG Cloud Survey Report,* s.l.: KPMG.

Krebs, R., Momm, C. & Kounev, S., 2014. Metrics and techniques for quantifying performance isolation in cloud environments.. *Science of Computer Programming,* Volume 90, pp. 116-134.

Krishnadas Nanath, R. P., 2013. A model for cost-benefit analysis of cloud computing.. *Journal of International Technology and Information Management,* 22(3), p. 6.

Lewis, G. A., 2013. *Role of standards in cloud-computing interoperability.*. Hawaii, IEEE, pp. 1652-1661.

Marc Walterbusch, B. M. F. T., 2013. Evaluating cloud computing services from a total cost of ownership perspective.. *Management Research Review,* 36(6), pp. 613-638.

Marston, S. et al., 2011. Cloud computing — The business perspective. *Decision Support Systems 51 (2011),* Issue 51, pp. 176-189.

Mell, P. & Grance, T., 2011. *"The NIST Definition of Cloud Computing" USA, National Institute of Standards and Technology.* [Online]
Available at: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[Accessed 07 May 2016].

Mohan Baruwal Chhetri, S. C. Q. B. V. R. K., 2013. *Smart CloudBench--Automated Performance Benchmarking of the Cloud.*. s.l., IEEE, pp. 414-421.

Natis, Y. V. et al., 2011. *Gartner - PaaS Road Map: A Continent Emerging.* [Online]
Available at: http://www.gxs.co.uk/wp-content/uploads/wp_gartner_paas_road_map.pdf
[Accessed 10 10 2016].

North Bridge, 2015. *2015 Future of Cloud Computing Study.* [Online]
Available at: http://wikibon.com/2015-future-of-cloud-computing-survey/
[Accessed 09 10 2016].

NorthBridge / Gigacom, 2014. *Cloud Computing Survey 2014.* [Online]
Available at: http://www.northbridge.com/industry-largest-cloud-computing-survey-reveals-5x-adoption-saas
[Accessed 1 10 2016].

Nuseibeh, H., 2011. *Adoption of Cloud Computing in Organizations.* s.l., AMCIS 2011 Proceedings.

PCI Security Standards, 2016. *PCI DSS v3.2.* [Online]
Available at: https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss
[Accessed 27 01 2017].

Phaphoom, N. et al., 2015. A survey study on major technical barriers affecting the decision to adopt cloud services. *Journal of Systems and Software,* Issue 103, pp. 167-181.

Privacy Rights Clearinghouse, 2016. *Privacy Rights Clearinghouse.* [Online]
Available at: https://www.privacyrights.org/data-breaches
[Accessed 09 10 2016].

Raja Appuswamy, C. G. D. N. O. H. A. R., 2013. Scale-up vs Scale-out for Hadoop: Time to rethink?. *Proceedings of the 4th annual Symposium on Cloud Computing*, 1 10, p. 20.

Repschlaeger, J., Wind, S., Zarnekow, R. & Turowski, K., 2012. *A Reference Guide to Cloud Computing Dimensions: Infrastructure as a Service Classification Framework.* Hawaii, IEEE.

Repschlaeger, J., Wind, S., Zarnekow, R. & Turowski, K., 2013. *Decision Model for Selecting a Cloud Provider: A Study of Service Model Decision Priorities.* s.l., AMCIS 2013 Proceedings.

Resources, Science, and Industry Division, 2002. *The Internet and the USA PATRIOT Act: Potential Implications for Electronic Privacy, Security, Commerce, and Government.* [Online]
Available at: https://epic.org/privacy/terrorism/usapatriot/RL31289.pdf
[Accessed 28 01 2017].

Rittinghouse, J. W. & Ransome, J. F., 2009. *Cloud Computing: Implementation, Management, and Security.* 1st ed. s.l.:CRC Press.

Riverbed Technology, 2013. *SAP Optimization Guidelines - An Analysis of Options for Optimizing SAP Business Applications with Riverbed.* [Online]
Available at: https://www.riverbed.com/content/dam/riverbed-www/en_US/Documents/fpo/Partners/Whitepaper%20-%20SAP%20Optimization%20Guidelines.pdf
[Accessed 22 10 2016].

Sadiq, M., Iqbal, M. S., Malip, A. & Othman, W. M., 2015. A Survey of Most Common Referred Automated Performance Testing Tools.. *ARPN Journal of Science and Technology,* 5(11), pp. 525-536.

Salapura, V. & Mahindru, R., 2016. Enabling enterprise-level workloads in the enterprise-class cloud. *IBM Journal of Research and Development,* 60(2-3), pp. 3:1-3:8.

Shay, L. A., 2016. *Aviation Week Network - Lufthansa Technik To Launch Big Data Analytics Platform.* [Online]
Available at: http://aviationweek.com/mro-europe-2016/lufthansa-technik-launch-big-data-analytics-platform
[Accessed 11 02 2017].

Stantchev, V., 2009. Performance evaluation of cloud computing offerings.. *Third International Conference on Advanced Engineering Computing and Applications in Sciences, 2009.*, 11 10, pp. 187-192.

Stratogen, 2015. *Cloud Adoption Survey 2015.* [Online]
Available at: http://www.accessalto.us.com/media/5511/hst-cloud-adoption-survey.pdf
[Accessed 26 01 2017].

Subhas Chandra Misra, A. M., 2011. Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment.. *Mathematical and Computer Modelling,* 53(3), pp. 504-521.

Talia, D., 2013. Toward cloud-based big-data analytics.. *IEEE Computer Science,* May, pp. 98-101.

U.S. Department of Health & Human Services, 1996. *HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT OF 1996.* [Online]
Available at: https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996
[Accessed 27 01 2017].

Varadarajan, V. et al., 2012. Resource-freeing attacks: improve your cloud performance (at your neighbor's expense).. *Proceedings of the 2012 ACM conference on Computer and communications security*, 16 10, pp. 281-292.

Vijay Janapa Reddi, B. C. L. T. C. K. V., 2010. Web search using mobile cores: quantifying and mitigating the price of efficiency.. *ACM SIGARCH Computer Architecture News,* 38(3), pp. 314-325.

Virtustream, 2014. *Business Critical Clouds - UK Market Survey.* [Online]
Available at:
http://www.virtustream.com/pdfs/Virtustream_WP_BusinessCriticalCloud_UKMarketSurvery.pdf
[Accessed 26 01 2017].

Wall Street Journal, 2014. *Are Consumers Better Off Putting Everything in the Cloud?.* [Online]
Available at: https://www.wsj.com/articles/are-consumers-better-off-putting-everything-in-the-cloud-1399644099
[Accessed 11 02 2017].

Ward, C. et al., 2010. *Workload Migration into Clouds Challenges, Experiences, Opportunities.* Miami, Florida, IEEE, pp. 164 - 171.

Zhining Wang, N. W., 2012. Knowledge sharing, innovation and firm performance.. *Expert Systems with Applications*, 11 March, 39(10), pp. 8899-8908.

Zhizhong Zhang, C. W. D. W. C., 2013. A survey on cloud interoperability: taxonomies, standards, and practice.. *ACM SIGMETRICS Performance Evaluation Review*, 40(4), pp. 13-22.