EBU7501: Cloud Computing
Week 1, Day 4: Cloud Scalability

Dr. Gokop Goteng

Lecture Aim and Outcome

- Aim
  - The aim of this lecture is to teach students cloud-based scalability with examples of hardware and software performance issues
- Outcome
  - After this lecture students should be able to:
    - Understand and implement the concept of Elasticity to make cloud systems scalable using Amazon AWS Elastic Compute Cloud (EC2)
    - Know types of scalability and performance issues
    - Use Amdahl's law and universal scalability law (USL) to determine the performance and scalability based on the ratios of parallel to serial processes in a cloud system

Lecture Outline

- Scalability
- Why Scale-Up a System?
- Different Measurements for Scalability
- Strategies to Implement Scalability
- Hardware, Software/Application and Database Scalability
- Performance and Hardware Scalability
- Universal Scalability Law (USL)
- Scalability in Cloud Computing
- Scalability using AWS Tools
- Class Task

Scalability

- The ability of a system to accommodate additional resources (software, hardware, database, functionalities) in order to handle more work load without major changes to the original setup and architecture
  - A computer system (hardware, software, database, network) that can maintain the same performance when additional users, data and computations are introduced is said to be a scalable system
    - Increase in usage, users, network traffic, computations, storage and data processing add pressure to computing resources which affect the performance of the system
    - Hence the need to implement scalable systems that can accommodate this increase in usage

Why Scale-Up a System?

- The ever increasing generation of data from many sources in research centres, universities, business community and social computing networks
- The globalisation of education and businesses that have branches around the world and need to integrate their computing systems for efficiency
- New technologies that require additional hardware and software
- Increase in business institutions through mergers, acquisitions and branches which may mean additional staff, offices, customers, data and computing systems
- In anticipation of future expansion and growth in an organisation
- Integrating public cloud and private or community cloud

Scalable Computing Trends

- Moore's Law (by Gordon Moore in 1965)
  - States that the number of components in integrated circuit will double every two years, In essence, it means that the processor speed will double every 2 years (agreed to be 18 months actually)
- Parallel Computing and Degree of Parallelism
  - Systems can improve scalability depending on the level of parallelism used
    - Bit-level parallelism (BLP), instruction-level parallelism (ILP), data-level parallelism (DLP)
    - DLP was made popular through the single instruction, multiple data (SIMD)
- Distributed and Cloud Computing
- Virtualisation and Virtual Infrastructures
- Cluster Computing
  - The main application areas for clustering are scalability and high availability

Computer Clusters for Scalability

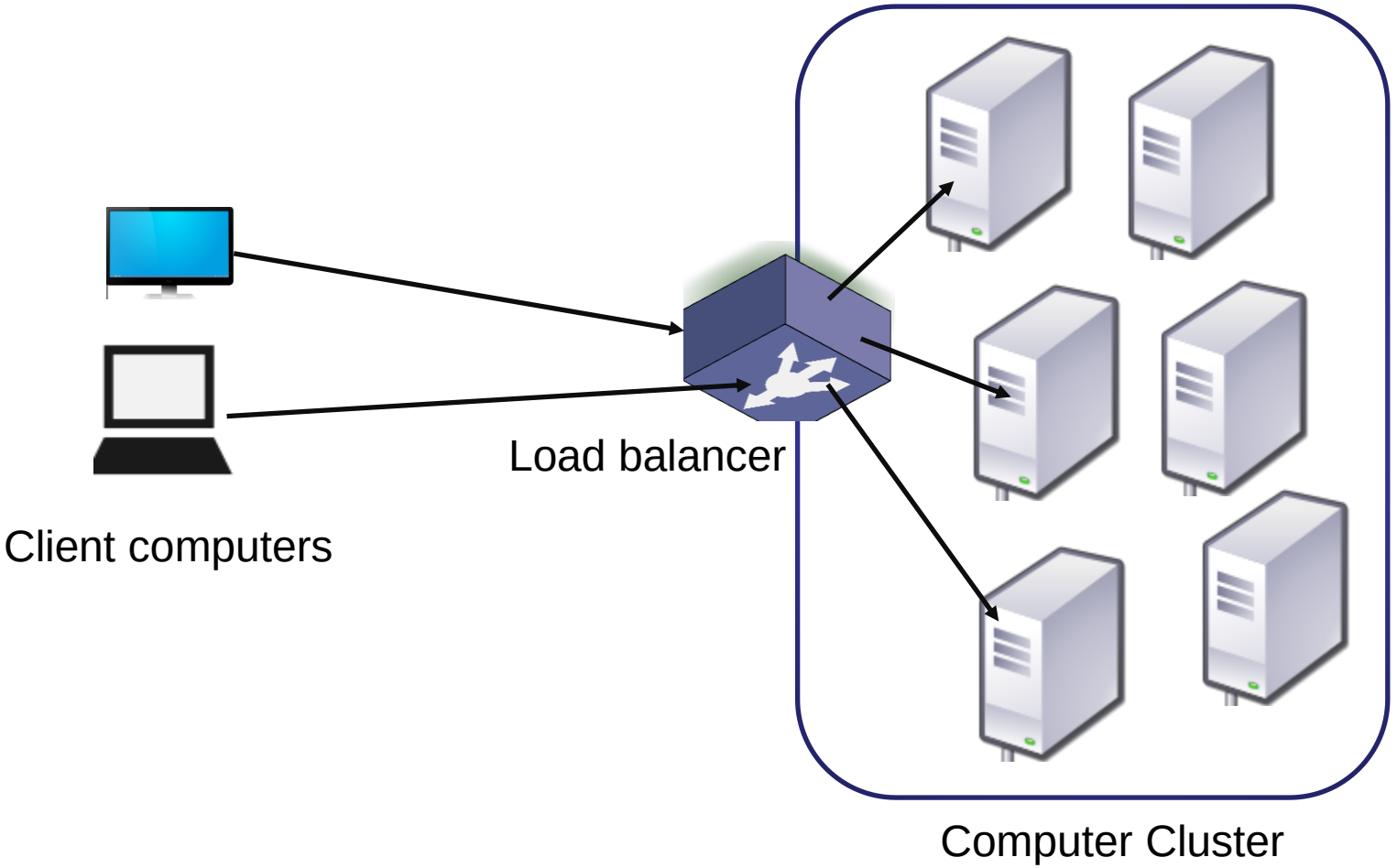
- A computer cluster is a collection of loosely or tightly coupled interconnected pool of computers that work together collectively and cooperatively as a single computing resource to solve the same or common task.
- Computer clusters are usually connected through a local area network (LAN)
- Clustering of computers enables scalable parallel and distributed computing as computation, data processing and additional functionalities and resources can be performed or added to the cluster to handle additional workload
- Clustering explores massive parallelism at the job level and achieves high availability (HA) through stand-alone operations
- The benefits of computer clusters and massively parallel processors (MPPs) include:
  - Scalable performance, HA, fault tolerance and modular growth

Computer Clusters for Scalability

- Scalable Design Objectives of Computer Clusters
  - Scalable modular growth: Clustering of computers is based on the concept of modular growth of the computational resources and network
  - For example we can scale (increase the size) of a cluster from hundreds of uniprocessor nodes to a supercluster of thousands of multicore nodes
    - This is a challenging task if all the nodes are physical nodes
    - The best way to do this is to use virtualisation technologies in a cloud environment, eg Amazon AWS EC2
  - Scalable performance: This means that scaling (adding) of cluster resources such as cluster nodes, memory capacity, I/O bandwidth, processors, disk storage, etc will lead to a proportional increase in performance and efficiency
    - In this regard, to ensure optimum utilisation of cluster resources, there should be scale-up and scale-down mechanisms to be used depending on the application and tasks you are carrying out.
      - In cloud systems, you can use dynamic scaling tools eg AWS Auto Scaling and Load Balancing
  - Clustering is primarily driven by scalability using parallel and distributed applications and high availability through fail over

Computer Clusters for Scalability

- Scalability through load balancing



Different Measurements for Scalability

- Scalability affects different parts of a computing system
- It is easy to scale up one part of a system while other parts are degrading in performance
- The following measurements of scalability highlights different parts of a system that need scaling up
  - Functional scalability
    - This is one of the common features of computing systems. Organisations are constantly adding new functionalities in their hardware and applications.
    - If additional functionalities are added and there is no degradation in memory usage, speed and latency, then the system is scalable
  - Geographical scalability
    - If a system can maintain its original performance both as a local with few local sites and distributed systems with multiple sites that are globally distributed, then it is said to be geographically scalable
  - Administrative scalability
    - If a system maintains the same performance after adding more users and organisations to the system, it is said to have administrative scalability

Different Measurements for Scalability

- The following measurements of scalability highlights different parts of a system that need scaling up
  - Heterogeneous and generational scalability
    - If adding new features and components (new generation components) that are from different vendors and manufacturers (heterogeneous) does not affect the performance of a system, then the system is said to exhibit heterogeneous and generational scalability
  - Load scalability
    - This is another common experiences that users of systems undergo. If an additional load (storage, computation, network traffic, etc) is introduced into a system and it still maintains its performance, then it said to have load scalability

Strategies to Implement Scalability

- Horizontal strategy
  - This strategy is used to add more computer nodes to an existing system to implement distributed system to enhance performance in distributed applications
- Vertical strategy
  - This strategy is to add additional computing resources such as memory, CPU, network cards to an existing single computer to enhance its performance. It could involve creating virtual machines on the same physical machine.

Hardware, Software/Application and Database Scalability

- Hardware scalability
  - This is the process of adding hardware such as a complete computer node, CPU, memory and network cards to scale a system
- Software/Application scalability
  - This is the process of adding functionalities and features to applications using APIs, plug-ins and libraries without affecting the performance
- Database scalability
  - This is the process of using techniques such as partitioning of database tables, multi-threaded implementations of database systems, network-attached storage (NAS) systems and storage area networks (SAN) to scale up database storage systems.
  - In addition, cloud systems are exploring NoSQL systems such as Casandra, Hadoop, Amazon's DynamoDB and Googles BigTable to scale up storage in cloud systems

Performance and Hardware Scalability

- Vertical and horizontal scaling of hardware enhance performance
- The performance is not directly proportional to the amount or number of resources added
  - This means that if for example 2 CPUs can process a particular task within 4 seconds, it does not necessarily mean that adding 2 more CPUs (to become 4 CPUs) will result in the same task completing within 2 seconds
- There is a threshold that the law of diminishing returns can set in
  - This is why users may consider performance tuning (changing parameters to speed up a system) when they notice diminishing returns in adding more resources

Performance and Hardware Scalability

- Amdahl's Law (also known as Amdahl's Argument)
  - Consider the execution of a given program on a uniprocessor workstation with total execution time of T minutes.
  - Let's assume the program is parallelised (partitioned) to execute parallel on a cluster of many processing nodes with ***n*** processors.
  - Assuming that a fraction, say **α**, of the program will be executed sequentially (called the ***sequential bottleneck***)
  - Therefore, (***1- α***) of the program can be executed in parallel
  - The total execution time of the program is then calculated by:
    - αT + (1- α)T/n
      - Where the first term ***αT*** is the sequential execution time on a single processor and the second term ***(1- α)T/n*** is the parallel execution time on n processing nodes
  - In this scenario, we assume that all communication overheads and I/O time are ignored
  - Using the assumptions above, Amdahl's Law states that the ***speedup factor***. S, of using the n-processor system over the use of a single processor:

$$\text{Speedup} = S = T/[\alpha T + (1- \alpha)T/n] = 1/[\alpha + (1 – \alpha)/n]$$

Where α is the fraction of the serial calculation (calculation done one after another in sequence), (1- α) is the fractional part that can be parallelised (multiple calculations that can be done at the same time) and n is the number of processors used

Performance and Hardware Scalability

- Amdahl's Law continued…
  - The maximum speedup of n processors is achieved only if the sequential bottleneck α, is reduced to zero or the code of the program is fully parallelisable with α=0
  - As the cluster becomes sufficiently large, meaning as n->∞, S approaches 1/ α, an upper bound on the speepup S.
  - For example, the maximum speed up is 4 when α=0.25 and (1- α)=0.75 even if one uses hundreds of processors
    - In this case increasing the size of the cluster alone may not result in a better speedup
  - This led Amdahl's Law to specify that the sequential portion should be as small as possible so as to get maximum speedup that is good enough as we increase the number of cluster nodes and processors

Performance and Hardware Scalability

- Amdahl's Law Continued…
  - Using this formulation for 2, 4, 8 processors if we assume that 60% of the task can be parallelised on multiple processors, leaving 40% of the task to be processed serially on a single processor you will get the following performances:
    - $1/(\alpha + ((1 - \alpha)/Np)) = 1/(0.4 + ((1 - 0.4)/2)) = 1.429$
    - $1/(\alpha + ((1 - \alpha)/Np)) = 1/(0.4 + ((1 - 0.4)/4)) = 1.818$
    - $1/(\alpha + ((1 - \alpha)/Np)) = 1/(0.4 + ((1 - 0.4)/8)) = 2.105$
  - Class Work – Use Amdahl's Law to do the following
    - Change the percentage to 50% each for serial and parallel calculations using (2, 4, 8 Processors)
    - Make 60% for serial and 40% for parallel calculations (2, 4, 8 Processors)
    - Draw on a graph your results including the 3 results above (2, 4, 8 Processors)
    - Discuss your observations based on scalability of hardware

Performance and Hardware Scalability

- Problem with Fixed Workload
  - In Amdahl's Law, the assumption is the use of same amount of workload for both sequential and parallel execution of the program with a fixed problem size or data set
  - Hwang and Xu called this fixed-workload speedup
  - To execute a fixed workload on n-processors, parallel processing lead to a system efficiency given as:
    - Efficiency = E = Speedup/n = S/n = $1/[\alpha n + 1- \alpha]$
  - The system efficiency is often very low especially when the cluster size is very large
  - If you execute the example of $\alpha=0.25$, $1- \alpha=0.75$ for n=256 cluster nodes, the efficiency will be:
    - $E=1/[0.25 \times 256 + 0.75] = 1.56\%$
  - This is because only a few processors (eg 4) are kept busy while the majority of the nodes (eg 252) are left idling.

Performance and Hardware Scalability

- Gustafson's Law
  - To achieve higher efficiency when using cluster, we must scale the problem size to match the cluster capability
  - To solve this problem, another speedup law known as Gustafson Law (or Scaled-workload speedup) was proposed by John Gustafson (1988)
  - Let's assume that W is the workload in a given program
  - When using an n-processor system, the user should scale the workload W to
    - **$W' = \alpha W + (1- \alpha)nW$**
  - Note that only the parallelisable portion of the workload is scaled n times in the second term

Performance and Hardware Scalability

- Gustafson's Law continued…
  - Gustafson's law states that scaled-workload speedup on a scaled workload W', is given as:
    - $S' = W'/W = [\alpha W + (1- \alpha)nW]/W = \alpha + (1- \alpha)n$
  - By fixing the parallel execution time at level W, the efficiency under Gustafson law becomes:
    - $E' = S'/n = \alpha/n + (1- \alpha)$
  - We can see that there is an improvement in the efficiency in the previous example as:
    - $E' = 0.25/256 + 0.75 = 0.751 = 75.1\%$
  - In summary, use Amdahl's law for a fixed workload, but use Gustafson's law for scaled problems

System Availability

- For systems to scale up well, high availability (HA) is required in clusters, parallel, distributed and cloud computing systems
- A system is said to be highly available if it has a long ***mean time to failure*** (MTTF) and a short ***mean time to repair*** (MTTR)
- System availability is given as:
  - System availability = SA = MTTF/(MTTF + MTTR)

Performance and Hardware Scalability

- Strong scaling
  - This shows how the performance (speed up time) varies with increase in the number of processors for a particular fixed size of the total problem
- Weak scaling
  - This shows how the performance (speed up time) varies with increase in the number of processors for a particular fixed size of a problem per processor

Scalability in Cloud Computing

- Elasticity of Computing Resources
  - The elasticity of cloud computing resources means that resources can be added or increased in number or capacity as well as removed or decreased in number or capacity depending on the user's service level agreement with the providers
  - The cloud offers flexible way to perform vertical scale up and scale down of resources on the fly as users use the system
  - Users can also perform horizontal scaling by adding multiple computing nodes to an existing systems within a very short period of time
- Virtualisation scaling
  - It is easy to create virtual machines to add to existing systems using cloud solutions
- Scaling through migration of resources
  - Cloud solutions offer tools to migrate virtual machines, data storage to a different geographical region to scale up the system when needed
- Load balancing
  - This provides capability to distribute job workloads to different nodes to optimise performance and optimum uitlisation of computing resources

Virtualisation Scaling and Performance Issues

- Virtualisation scaling saves time and cost of adding large cloud resources within a short time
- However, this may affect the performance of resources compared to the hardware components
- Memory virtualisation creates memory but with less performance in terms of storage rate
- Processor virtualisation also creates virtual processors that perform computation and processes data at a slower rate than hardware processors

Large-scale Systems Requires Self-Organisation

- AWS EC2 has about 1 million servers in 2015
- Google has about 3 million servers in 2015 to run its cloud and search engines
- The number of servers keep increasing at a rapid rate
- This requires intelligent and automated way of scaling up the number of servers
- The complexity of interactions among servers calls for self –organisation of the servers and instances
- The most important attribute of self-organisation is scalability

Scalability using AWS Tools

- Amazon AWS Auto Scaling
  - Scales up Amazon Elastic Compute Cloud (EC2) automatically based on the configurations defined in users set up
  - It can be set up so that the number of EC2 can increase automatically depending on the load threshold defined by the user
  - It scales with load balancing
  - Read more at http://aws.amazon.com/autoscaling/
- Amazon AWS CloudFront
  - AWS CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds regardless of the location of users.
  - CloudFront uses the AWS Edge Locations globally to cache contents (data, videos, etc) using Edge Caching which allows content to be served by infrastructure that is closer to viewers, lowering latency and giving you the high, sustained data transfer rates needed to deliver large popular objects to end users at scale.

Scalability using AWS Tools…

- AWS Load Balancing
  - Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.
  - It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.
- Use Amazon EC2 and S3 (simple storage service) APIs to scale your cloud systems to private or community cloud
  - Creating a hybrid cloud system
- Use Amazon APIs to scale to open cloud systems such as OpenStack (Rackspace)

Scaling Across Regions and Zones

- AWS scales its servers and applications across Regions and Zones around the world
- A Region is a physical geographical location which consists of one or more Zones
  - Eg Europe is an AWS Region and it consists of Zones such as London, Ireland, Paris, etc and China Region consists of Beijing and Ningxia Zones
  - Availability Zones increase reliability, availability and throughput
- Elastic Load Balancer and Auto Scaling automatically handles the deployment, capacity provisioning and load balancing as a function across regions and zones
- Automatic migration of servers to scale up computations and data storage

Class Task

- What is AWS CloudFront?
- What are AWS Regions and Zones?
- What is AWS Edge Location and Edge Caching?
- Use Amdahl's Law to draw the graph of a system with 70%, 80% and 95% parallel processes running on 4, 8, 16 and 32 processors
- Reverse the percentages of 70%, 80% and 95% to represent serial processes with the same number of processors respectively
- Compare the 2 graphs and make your conclusions in terms of hardware scalability

Class Work

Scalability is an important feature in cloud computing systems.

- Briefly explain scalability in terms of computer systems **[2 marks]**
- Differentiate between horizontal and vertical scalability **[2 marks]**
-

Class Work

To provide cloud services, cloud providers use cloud solutions:

- List TWO commercial cloud solutions **[2 marks]**
- List TWO open source cloud solutions . **[2 marks]**
- Describe the function of a Hypervisor (or Virtual Machine Manager) **[1 mark]**

Class Work

- Define Virtualisation and list TWO advantages of virtualisation. **[4 marks]**

Class Work

Service level agreements (SLAs) promote quality of cloud services provided to customers.

- Describe Quality of Service (QoS) in terms of cloud computing **[2 mark]**
- ii. List TWO cloud-based parameters for measuring QoS **[2 marks]**

Class Work

Amdahl's Law (also known as Amdahl's Argument) describes the speedup in parallel computing systems when more processors are added to a system. Using this knowledge of Amdahl's Law: **[9 marks]**

- State Amdahl's equation: **(3 marks)**
- Explain what **α** and **Np** stand for in Amdahl's Law **(2 marks)**
- Using Amdahl's Law, calculate the performance that can be achieved for 4 and 8 processors if **80%** of the task is done in parallel **(2 marks)**
- Explain the results you obtained when more CPUs were added. **(2 marks)**