International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 30 No. 1 Jul. 2020, pp. 199-214

http://www.ijias.issr-journals.org/

The past, present and future of scalable computing technologies trends and paradigms: A survey

Amany Abdelsamea, Salwa M. Nassar, and Hesham Eldeeb

Computers and Systems Department, Electronics Research Institute, Giza, Egypt

ABSTRACT: The past era saw a significant boost in computing, storage, and networking technologies which caused the development of massive-scale adaptive applications in engineering and science. These complex, dynamic and heterogeneous applications are combined with correspondingly heterogeneous and complex distributed and parallel computing systems caused the improvement and implementation of efficient computational infrastructures which allow execution, programming and runtime management support for these massive-scale adaptive implementations. In this paper different types of computing technologies are described. In fact, all these technologies have contributed in the development of computing. Determining one specific technology as the best among others is very difficult, since every day is an evolution in computing and every single paves a way for a new technology. This paper presents a comprehensive review for the past, present and future of scalable computing technologies trends and paradigms. Firstly, high performance computing technologies are presented. This paper presents a new classification of high performance computing into Supercomputing and Quantum Computing where Supercomputing is classified into petascale, exascale and zettascale computing. The paper discusses the main challenges in exascale computing and quantum computing and a comparison between classical supercomputing and quantum computing is presented. Secondly, distributed computing technologies specifically Peer-to-Peer Computing, Cluster Computing, Grid Computing and Cloud Computing are presented. The paper discusses their advantages and disadvantages and a comparison between them is also presented. Thirdly, post Cloud Computing Paradigms mainly dew, mist, edge and fog computing are presented. Fourthly, Jungle computing is presented. Finally, the paper highlights that exascale and quantum computing are the most recent topic to effectively achieve high performance computing, both technologies have their advantages and disadvantages so it is recommended to implement a hybrid system that uses both technologies so quantum computing can be used as an accelerators to the existing high performance computing systems. Supercomputers have a very high cost so distributed computing systems that provide high performance along with versatility and cost efficiencies are developed. Understanding and utilizing post cloud computing technologies correctly with cloud computing can help in IOT solutions.

KEYWORDS: High Performance Computing, Quantum Computing, Exascale Computing, Distributed Computing and Cloud Computing.

1 INTRODUCTION

Computing describes how computers and computer systems operate and how they are designed and programmed [1]. The computing paradigm changed and developed with the advancement of computer hardware, network and software technology. Historically, the computational model evolved from mainframes pre 1970 to Personal Computing (PCs) in 1980 as seen in Fig. 1. With the development of networking, personal computing then evolved to cluster computing [2] in 1993 and grid computing [3] in 2000. In network computing era, cloud computing [4] is commonly perceived as the dominant paradigm in 2010. With the fast advancement of ubiquitous network technologies and pervasive intelligent devices, unique network applications are developing, for example, the Internet of Things, smart grids, smart cities, unmanned vehicles and virtual/augmented reality.

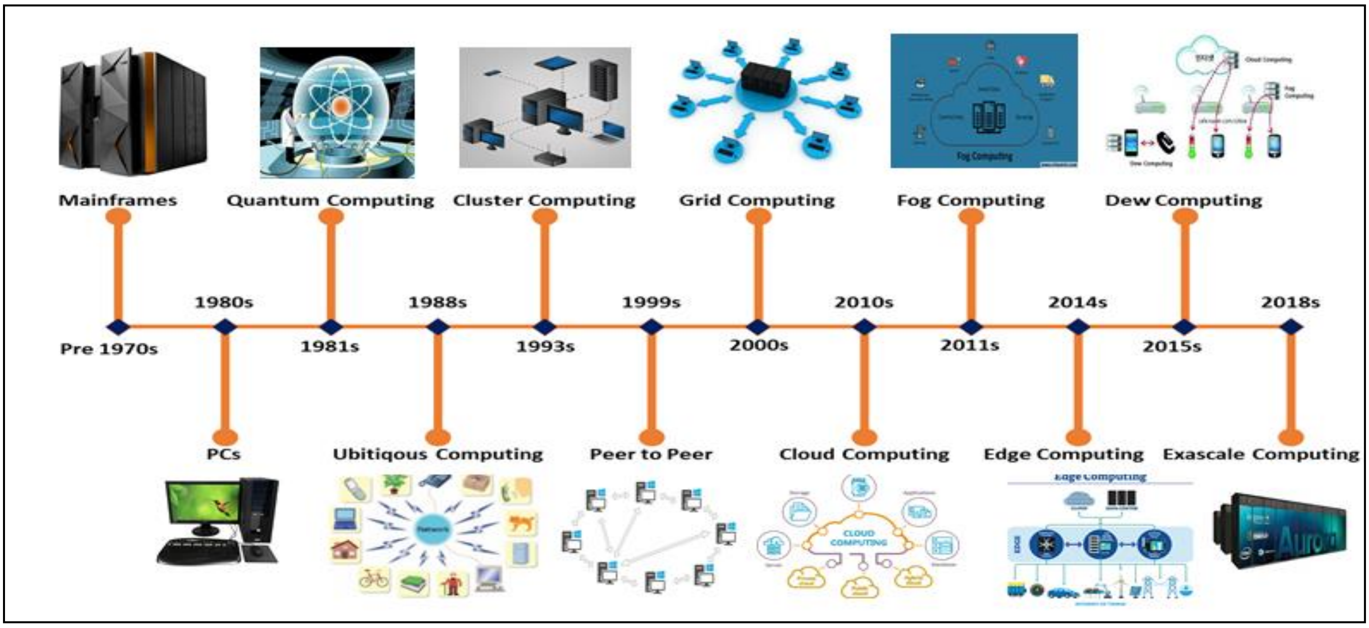

Fig. 1. Computing TimeLine

Cloud computing is experiencing issues addressing the requirements of those developing applications and technologies due to its centralized computation and storage. To overcome these issues, new types of computing have been developed to expand the cloud computing to the edge of an enterprises network. These computing types include fog computing [5] in 2011, mobile edge computing [6] in 2014 and dew computing [7] in 2015. High-performance computing (HPC) [8] is crucial for scientific and engineering applications that request advanced computation. The performance of HPC systems improves with the development of the semiconductor technology, that follows Moore' s law [9] stating that the number of transistors integrated into a chip will approximately double each 24 months. With Moore's law moving toward its point of confinement, we will confront a lot more extraordinary difficulties than any other time in realizing a sustainable exascale [10] and zettascale [11] computing systems beyond 2018. These challenges arise due to process limit, communication, power consumption, storage, memory, programming and reliability. All the previous computing technologies utilize classical computing but nature is not classical, dammit (Richard Feynman 1918-1988), also by 2020 to 2025, transistors will be so small and will produce so much heat that standard silicon technology may eventually collapse so quantum computing [12] has now compelled to move from theory to reality. A classification of these computing technologies is presented in the next subsection.

1.1 Computing Technologies Classification

There has been a massive exponential growth in computer processing power, networking and data storage over the past decade. Accordingly there has been a major change in computation from earlier times until today. Computing technologies are classified into High Performance Computing (HPC), Distributed computing and Jungle computing technologies as shown in Fig. 2. The high performance computing technologies are classified into Supercomputing and Quantum Computing. Supercomputing is classified into Petascale computing, Exascale computing and Zettascale computing.

The distributed computing technologies are classified into Peer-to-Peer computing, Cluster computing, Grid computing and Cloud computing. A lot of technologies arise that extend and utilize cloud computing. These technologies are Fog computing, Mobile edge computing, Mist computing and Dew computing. This paper describes the past, present and future of scalable computing technologies trends and paradigms. A full description and comparison between these technologies is presented in the next sections.

The paper is set out in the following way: Section 2 explains high performance computing technologies. Section 3 discusses the distributed computing technologies. Section 4 explains Post Cloud Computing Paradigms. Section 5 discusses the Jungle computing technology. Finally, the conclusion and future work is discussed in section 6.

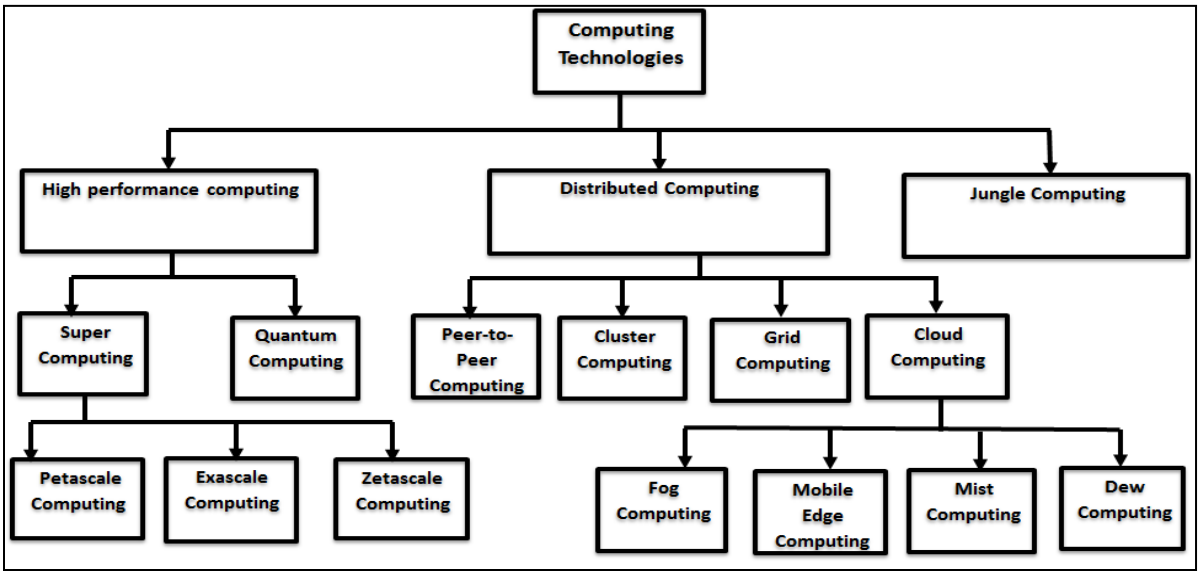

Fig. 2. Computing Technologies Classification

2 HIGH PERFORMANCE COMPUTING

High Performance Computing (HPC) [8] corresponds to the utilization of supercomputers and massively parallel processing techniques to handle complex computational issues using computer simulation, modeling and data analysis. High performance computing combines many technologies, involving computer architecture, application software, electronics, programs and algorithms under a solitary framework to handle advanced issues rapidly and adequately. High Performance Computing is especially appropriate to tasks which are either numerically, computationally or data intensive along with tasks which involve significant number of complex computations to be accomplished rapidly on huge data sets. This makes high performance computing effective in all compute-intensive research fields especially biology, physics, earth sciences, climate modeling, national security, aerospace, engineering and energy. High Performance Computing is classified into Supercomputing and Quantum Computing.

2.1 SUPERCOMPUTING

Supercomputing [13] corresponds to the processing of hugely complex or data-laden tasks utilizing the concentrated compute resources of various computer systems working in parallel (i.e. a ”supercomputer”). Supercomputing allows problem solving and data analysis which could be basically impossible, very time consuming or costly with standard computers. The speed of computers is estimated by their ability to calculate floating point operations per second (or flops). Supercomputing requires a system that works at the greatest potential performance of any computer, regularly estimated in petaflops.

A supercomputer system is certainly a network of CPUs (e.g., microprocessors) which has numerous computing cores besides its own local memory to run a broad scope of software programs. The software programs which coders develop to operate on supercomputers are divided into several smaller independent computational processes, called threads, which can be performed concurrently on these cores. For supercomputers to function properly, their core needs to be well designed to communicate data effectively. Modern supercomputers contain of more than 100,000 cores or more. Exponential increase of supercomputing capacity as reported in the TOP500 list of supercomputers [14] is shown in Fig. 3.

The worlds´ fastest supercomputers currently solve problems at the petascale which is a quadrillion ($1* 10^{15}$) 1000 trillion FLOPS. In the last decade, the first petascale supercomputer was built in 2008. Fig. 4 shows the pre-exascale and exascale systems. Mira, Titan and Sequoia supercomputers appeared in 2013. Mira is known to be one of the fastest supercomputers which are ten petaflops IBM BlueGene/Q system. Mira provides ten quadrillion calculations per second. With this computing power, Mira can perform in one day what it will take a personal computer twenty years to accomplish. Titan is also a Cray hybrid architecture system that provides 27 thousands trillion calculations per second (27 petaflops). Sequoia is also one of the most powerful supercomputers in the world. The system provides two operations which are calculating the uncertainties in numerical simulations of nuclear weapons performance and conduct the advanced weapons science calculations required to build reliable physics based models for weapons codes.

Theta, Cori and Trinty supercomputers are established in 2016. Theta is an 11.69 petaflops system that relies on the second-generation Intel XeonPhi processor allowing a breakthrough in engineering research and computational science. Cori, the Cray

XC40 system is the recent addition to the National Energy Research Scientific Computing Centers (NERSC) supercomputing repertoire. It provides 30 Pflop/s. It supports scientific simulations and data-intensive workflows. Trinity, a Cray XC40 supercomputer provides 20.16 petaflops.

Summit 200 petaflops is the next jump in leadership class computing systems for open science. Summit, announced in 2018, performs 8 times the computing performance of Titans 18,688 nodes, by just using 4,608 nodes. Summit has hybrid architecture as Titan, and every node has various IBM POWER9 CPUs and NVIDIA Volta GPUs both associated together with NVIDIAs high-speed NVLink. The IBM Sierra supercomputer offers 4 to 6 times the sustained performance and 5 to 7 times the workload performance of Sequoia, with a 125 petaFLOP/s peak. At approximately 11 megawatts, Sierra is considered 5 times more powerful than Sequoia. Sierra incorporates 2 kinds of processor chips IBM Power 9 processors and NVIDIAs Volta graphics processing units (GPUs) so it is a promising architecture for extreme scale computing.

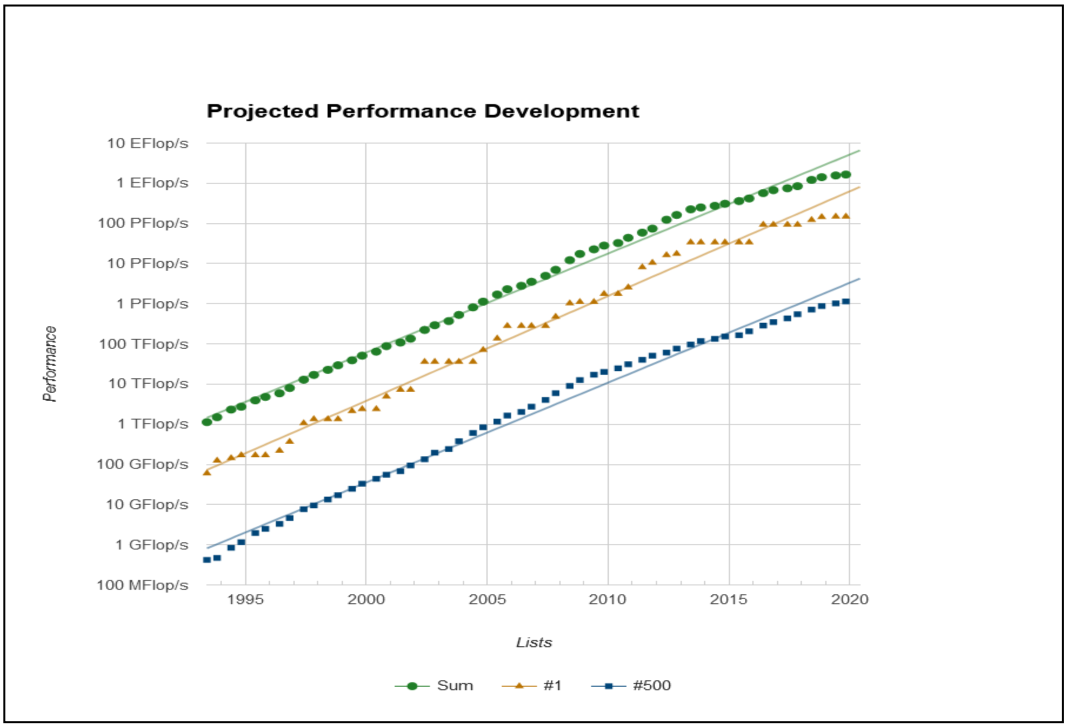

Fig. 3. Projected Performance Development as recorded by the TOP500 list [8]

Perlmutter (NERSC-9) 100 petaflops supercomputer is the successor to Cori, a planned supercomputer expected to be run by the National Energy Research Scientific Computing Center (NERSC). It is estimated that Perlmutter will be delivered in late 2020. By 2021, the goal of running high resolution 3D simulations with full physics and geometric features will be achieved using Crossroads supercomputers.

Although these petascale systems are very efficient, the next breakthrough in computing achievement is the exascale ($1 * 10^{18}$ floating point operations/s) a higher degree of performance in computing that will have a huge effects on daily life. Exascale supercomputers [10] will fastly analyze huge volumes of data and more practically simulate the complex tasks and interactions behind most of the fundamental forces of the universe. Exascale ($1 * 10^{18}$ operations/s) computing systems are scheduled to be implemented in 2021-2022. The supercomputer "Aurora" is considered as the first exascale device in the United States. Aurora is actually named as the third cornerstone of the CORAL pre-exascale project. It is developed by Intel and Cray for Argonne National Laboratory, however the date of delivery has moved from 2018 to 2021, also the overall capacity is increased from 180 petaflops to 1,000 petaflops (1 exaflop). Frontier is another exascale supercomputer that scheduled for delivery in 2021; Frontier will speed up innovation in science and technology and sustain the US leadership in artificial intelligence and high-performance computing. Frontier integrates artificial intelligence with data analytics and modeling and simulation. ElCapitan exascale supercomputer can have a peak performance of over 1.5 exaflops also an estimated launch at the end of 2022.

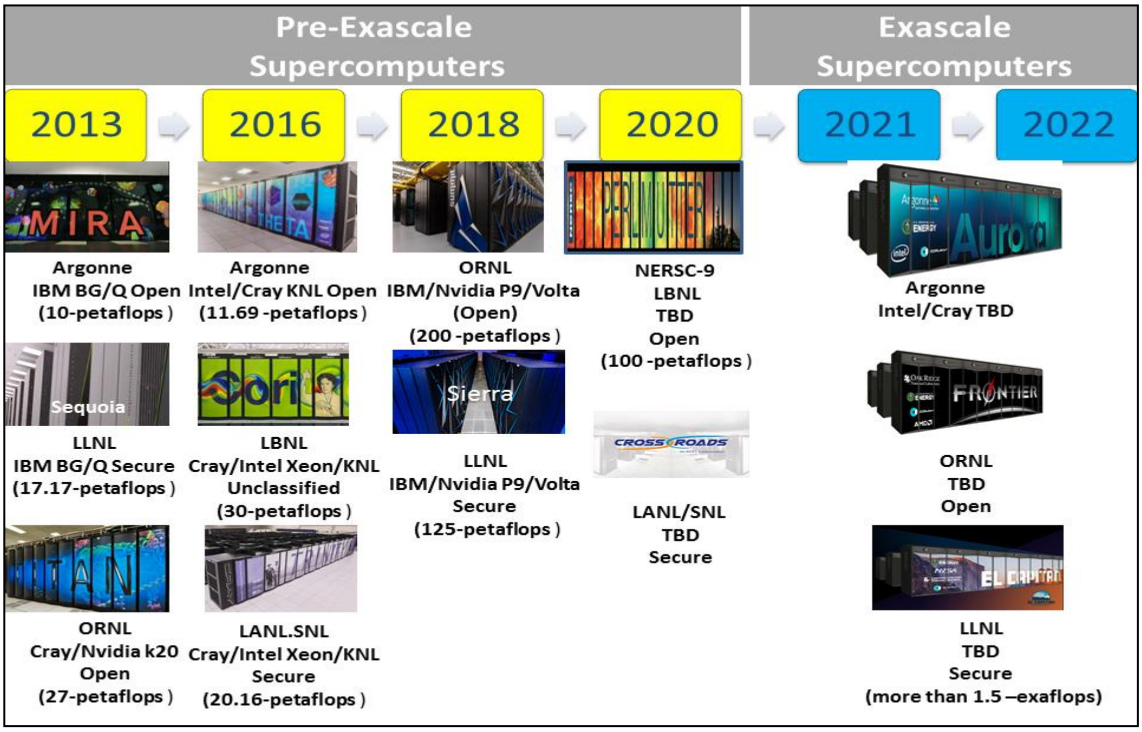

Fig. 4. Exascale TimeLine

The following are the main challenges in exascale computing [15]:

- Power and Cost are the major challenges. There will be a need for new architectures to fix the current energy consumption problems.
- Hardware failure hazards increase as new supercomputers includes a greater number of components. It may help to restart programs automatically on failure.
- It will also be necessary to write software for the new architecture which becomes more complex as accelerators are associated.

While the speed of performance increase slows down, the High performance computing systems are likely to enter the next milestone, zettascale computing ($1*10^{21}$ operations/s)*, by 2035 [16]. The search for technology paths that allow performance beyond exascale may need alternative computing models such as quantum computing since several problems are ideally suited to computational models other than the standard Turing machine model.

2.2 Quantum Computing

Quantum physics is a basic physics theory which identifies nature at the smallest scales of energy levels of atoms and subatomic particles. Quantum Computing relates the characteristics of quantum physics to process data. Quantum Computing functions with nanoscale devices at temperatures colder than intergalactic space. Quantum computing is not a new idea since it was introduced as a revolutionary technique in the early 1980s to address complex issues exponentially faster than classical computing can. Feynman (1982) introduced the idea of developing machines based on quantum mechanics laws rather than the laws of classical physics. The quantum turing machine was invented by David Deutsch (1985), demonstrating that quantum circuits become universal. Peter Shor (1994) developed a quantum algorithm to factor enormous numbers in polynomial time. Lov Grover (1997) establishes a quantum search algorithm with *O (sqrt (N))* complexity.

Among the most significant aspects of quantum computing is the quantum bit (Qubit), a unit of quantum information which occurs in two states (horizontal and vertical polarization) simultaneously due to the superposition concept of quantum physics. Quantum Computers utilize the quantum superposition, entanglement, tunneling and annealing to process the information. Quantum Superposition is the quantum mechanics building block which adds two quantum states together to obtain another quantum state that will be true or vice versa. Superposition demonstrates the wave like status and applies Schrodinger Wave equation. Quantum Entanglement is a situation whenever two or more particles are formed so that the quantum state of a particle will not be destroyed independently. Regardless how far the particles are they can reflect same quantum state. Every modification toward one end in the quantum state will indeed be liable on the change on the other particle.

A Comparison between Classical Supercomputing and Quantum Computing is presented in Table 1. Conventional computers use bits, that are restricted to take one out of two values, 0 or 1. Quantum computers, on the other hand, work with quantum bits, Qubits, which operate simultaneously with the superposition of both states. Quantum superposition makes quantum computation a special character with novel quantum gates that give rise to very efficient quantum algorithms but in classical computing the information Processing is performed by logic gates.

A three-bit register will generate eight choices 000, 001, 010, 011, 100, 101, 110, 111 using classical bits. Only one of these eight values will be taken by any classical record. On the other hand, if there is a three-qubit register, the system holds information about the eight different values at the same time due to quantum superposition. Therefore, a three-qubit register enables operations on eight alternatives in parallel. The number of operations performed is, in general, exponential in terms of the number of qubits. Therefore, depending on the number of qubits, a quantum machine is more or less effective. For fewer qubits, a quantum computer could not tackle very complex issues but will double its classically equivalent processing power with each additional qubit. Quantum computers provide an advantage in computational power over classical computers and they also use much less energy than classical machines.

Quantum computers are supposed to work with traditional computers. A Quantum Processing Unit (QPU) will take over to support the classical CPU whenever a task is too heavy for a classical computer. Since both supercomputing and quantum computing have their ad- vantages and disadvantages so it is recommended to implement a hybrid system that uses both technologies. Also quantum computing can be used as an accelerator to the existing HPC systems. Since Supercomputing is very costly so distributed computing systems that provide high performance along with versatility and cost efficiencies are developed.

Table 1. A comparison between classical Supercomputing and Quantum computing

Comparison	Classical Supercomputing	Quantum Computing
Information Storage	Bit based on voltage or charge	Quantum bit based on the direction of an electron spin
Information Processing	Achieved by logic gates e.g. NOT, AND, OR etc.	Achieved by Quantum logic gates
Circuit Behavior	It is governed by classical physics	It is governed explicitly by quantum mechanics
Representing Information	Utilize binary codes i.e. bits 0 or 1 to represent information	Utilize Qubits i.e. 0, 1 and both of them simultaneously to run machines quicker.
Operations Definition	Represented by Boolean Algebra	Represented by linear algebra over Hilbert Space
Advantages	- There are no limitations on copying or measuring signals. - Circuits are easily integrated in quick, scalable and macroscopic technologies such as CMOS.	- Faster than classical computation - Quantum computer can tackle classes of problems that choke conventional machines - The use of quantum computing is very green in nature. It can save enormous heat consumption in datacenters
Disadvantages	- Slower than Quantum computation	- Severe restrictions occur on measuring and copying signals - Circuits should utilize microscopic technologies which are not scalable, slow and fragile e.g. Nuclear magnetic resonance - High Q Error rates - Qubits live very short - Physical size of the machines is too huge to be of practical use to everyday society

3 Distributed Computing

Distributed Computing [17] is a type of computing where data and applications are distributed across different systems, but are linked and interconnected via network services and standards of interoperability so that they operate as a single environment. Distributed computing is classified into peer-to-peer computing, cluster computing, grid computing and cloud computing.

3.1 PEER-TO-PEER COMPUTING

Peer-to-peer (P2P) computing [18] is a computer networks using a distributed architecture. In P2P networks, computers devices are pointed to as peers, and they interchange and swap workloads. Every peer is equivalent to the other peers. There aren't any privileged peers, and there is no principal administrator device in the middle of the network. Peers are both clients and servers simultaneously. The most widely recognized application for peer-to-peer networks is the sharing of files on the internet since they permit the devices associated with them to get files and send files at the same time. Without the need for a central server, files can be exchanged directly between network systems. In other words, every computer on a Peer-to-peer network becomes both a file server and a client.

An Internet connection and Peer-to-peer software are the only criteria for a device to join a P2P network. The most popular Peer-to-peer software programs are Kazaa, BearShare, LimeWire, Morpheus and Acquisition. Such programs link to Peer-to-peer network, like"Gnutella, " that enables the computer to access thousands of other computers on the network. When associated with the network, P2P software lets you check for files on others computers. Additionally, other clients on the network can look for files on your computer, however normally just inside only a single folder, that you have allocated to share. As nodes in peer to peer networks operate as both clients and servers, it is hard to give appropriate security for the nodes. This may lead to denial of service attacks.

Some advantages of peer to peer computing [19] are as follows:

The network is very easy to deploy and manage as each computer in the P2P network manages itself.

- In the client server network, manages all the application requests of the clients. In peer-to-peer computing this requirement is not needed and the server's expense is saved.
- Scaling the peer to peer network and adding more nodes is easy. This only improves the data sharing capacity of the system.
- Disadvantages of Peer to Peer computing [20] are as follows:
- Backup of the data is complicated as it is stored in various computer systems also there isn't any central server.
- It is impossible to satisfy overall security in the P2P network since each system is independent and includes its own data.

3.2 CLUSTER COMPUTING

The difficulty and size of the present generation of supercomputers contribute to the advancement of cluster computing that is identified by its scalability, configuration and upgrade versatility, high availability and cost and time enhancement. Clusters were developed as a replacement to the costly multiprocessor supercomputer. Cluster computing [21] is a set of individual computers that are connected through specialized hardware and software, presenting a single system illusion to its users. In cluster computing environment the users specify the resources that is needed to run their application. Clusters can be classified according to the hardware into workstation based clusters which are composed of cheap computers and high performance clusters which use supercomputers to solve advanced computational problems. Cluster computing offers several important advantages [22]:

- Cluster computing may scale up to very huge systems.
- Improved price/performance ratios.
- High availability clusters offer various repetitive identical resources that, whenever managed properly, will provide continuous operation of the system even if individual components fail.
- Configuration and upgrade are flexible.

While clusters have many advantages, they also have disadvantages [23]:

- Network hardware isn't configured for parallel processing. Usually latency is so high and bandwidth is extremely low comparable to Symmetric Multiprocessor (SMP).
- Small computing power usage rate since the cluster nodes are not used completely most of the time.

3.3 GRID COMPUTING

Grid computing [24] is the process by which distributed heterogeneous computing resources can be used to solve computational problems so that ideal resources can be used effectively. This allows users to have on-demand computing. In a grid environment the resources do not have prior information about each other. Computing is ubiquitous in grid computing and individual users acquire accessibility to computing resources (processors, storage, applications, data, and so on) as required with no information of where these resources are placed and what the associated technologies, hardware, operating system, etc. are used.

A grid consists of various kinds of resources owned by different and usually independent organizations, resulting in resource and policy heterogeneity [25]. So, grid based services and applications encounter a different resource behavior than expected. Grids concentrated on incorporating existing resources with their operating systems, hardware, security infrastructure and local resource management. The support of the creation of Virtual Organizations which are logical entities within which distributed resources will be identified and shared as if they were in the same organization. Grids produce a group of toolkits, middleware, services and standard protocols. Security and interoperability considered as the main goals for the infrastructure of the grid as resources arise from different administrative domains, that have local and global resource utilization techniques, various hardware and software configurations and platforms, and differ in capacity and availability. There are no official standards for grid architecture but Globus Toolkit has evolved as the de-facto standard for many major fabric, core middleware, user level middleware and grid portals and applications.

Advantages of Grid Computing are [26]:

- Easy to collaborate with other organizations.
- Make better utilization of existing hardware.
- No need to buy huge servers for applications that can be split up and farmed out to smaller commodity type servers.
- Grid environments are much more modular and don't have single points of failure. If one of the servers/desktops within the grid fails there are plenty of other resources able to pick the load.
- Jobs can be executed in parallel speeding performance.
- Can solve larger, more complex problems in a shorter time.

Disadvantages of Grid Computing are [17]:

- Requires Fast Interconnection between compute resources (gigabit ethernet at a minimum) and Infiniband for MPI intensive applications
- Some applications need Customization. Applications will require tweaking to take full advantage of new models.
- Licensing across several servers will make it prohibitive for some applications.

Another technology developed after the Grid is the utility computing [27] that allows resource provisioning easier and on demand of user. Utility computing is a service provisioning paradigm where a service provider offers computing resources and infrastructure management to the clients as needed, and allows them to pay based on their particular usage rather than a flat rate. The utility model seeks to improve the efficient resources utilization as well as the reduction of the costs. At that point, we finally got the idea of cloud computing that concentrates on the provisioning and deprovisioning of computation, storage, data services to and from the user.

3.4 Cloud Computing

Cloud computing [28] is quickly growing as an alternative to traditional computing. Cloud computing concept is a development trend dependent on many successful researches in computing areas like virtualization, distributed computing, cluster computing and grid computing [29]. The term Cloud Computing came into light in 2006, after Amazons Elastic Computing Cloud (EC2) enters the world [30]. As per the National Institute of Standards and Technology (NIST) [31] Cloud Computing is defined as:

”Cloud computing is a paradigm for allowing on- demand, flexible network access to a shared pool of configurable computational resources which may be fastly provisioned and released with reduced management effort or service provider intervention”.

Cloud layout consists of five main features:

On-demand self-service: The user may provision computing resources automatically when desired without demanding human interaction with each service provider.

Broad network access: Resources are accessible over the network through standard techniques that improve usage by heterogeneous thin or thick client platforms (e.g., mobile phones, Personal- Digital-Assistants (PDAs) and laptops).

Resource pooling: The computational resources are grouped to fulfill diverse end-users needs with a multi-tenant model, with diverse virtual and physical resources continuously reassigned depended on user’s needs.

Rapid elasticity: Resources will be fastly provisioned automatically in some situations, to fastly scale out and quickly released to scale in.

Measured Service: Cloud computing frameworks regulate and maximize resource usage automatically by utilizing a monitoring and metering tool at some level of abstraction according to the kind of service.

NIST divides cloud computing into three as-a-service offerings, namely software, infrastructure and platform. Cloud computing is providing everything-as-a-service (XaaS) [32].The broad variety of services known as the Cloud computing stack can be arranged and composed into three large offerings that are accessible to end users. These are: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Software as a Service (SaaS): In SaaS, you have access to the interface of the software you need. You only utilize the software when you need. The software does not have to be installed on your machine; it resides on the server of the cloud service provider which also handles its support and maintenance.

Platform as a Service (PaaS): The provider of cloud service controls the operating system, data virtualization, servers, data storage, and networking in PaaS. The customer handles the applications which he uses.

Infrastructure as a Service (IaaS): It is also named Hardware as a Service (HaaS). In this service, you outsource the storage, hardware, servers, and networking components that you need from the cloud service provider. Nevertheless, it is the user's responsibility to manage the operating system, middleware and data.

Depending upon the service provider there are different types of clouds, each with its own advantages and disadvantages [33]. For example, one class of service providers is paying attention in reducing operating expenses, but others can concentrate on high reliability and security. Different possible models are Public Cloud, Private Cloud, Hybrid cloud and Community cloud.

Private cloud: The infrastructure of the cloud is performed specifically for an organization. It can be regulated by the organization or a third party also it may occur on premise or off premise.

Community cloud: Many organizations share the cloud infrastructure and support a specific community with shared priorities (e.g., mission, security requirements, policy, and compliance considerations).

Public cloud: The infrastructure of the cloud is accessible to the general public or a big industry association and is owned by a cloud service sale company.

Hybrid cloud: The infrastructure of the cloud consists of two clouds or more (public, private or community) which is unique entities but they are connected by standardized or customized technology which allows data and application portability.

Cloud computing advantages are:

- Pay-per-use, only as long as you need.
- Extensive storage capacity because the data is stored on the service provider's servers. Cloud service providers provide unlimited storage of data.
- Facilitated backup and recovery: For cloud computing, you can archive your data on a cloud-based server rather than a hard disk or a tape. Cloud service providers also allow dependable data recovery. This makes it easy and secure to back up your data.
- Cloud services can be accessed from anywhere, anytime using any device (e.g. a personal computer or a smartphone) connected to the Internet.
- Services can be set up easily and quickly.
- Easily scalable
- Instant software updates.

Disadvantages of cloud computing are:

- Lost control arises with handling over your data and information
- Based on third-party to guarantee the security and confidentiality of data and information
- Technical problems and network issues: There will be periods when the system may not operate well, due to failures, downtime, and other technical problems.

Another problem that cloud computing often doesn't handle well is time delay. Since data can be processed only in the cloud, there is always some delay between recording the data and getting results from that data.

Peer to peer, cluster, grid computing and cloud computing give some common characteristics for per- forming service-oriented or utility computing although peripheral variations occur on their procedures and methodologies from multiple perspectives as shown in Table 2. Some parameters used include: ownership type, virtualization etc. Their features are related though there are small differences. The whole model provides some degree of virtualization and scalable sizes. Al- though, they are different in many respects in terms of resource management, multi-tenancy, self-service and standardization. Cluster computing resources are administratively positioned and handled in a single do- main whereas grid resources are spread across multiple geographically administrative domains with their own objective management policies and goals. Cloud computing technology, on the other hand, incorporates major features of both cluster and grid computing, besides its own features and capabilities as significant support for virtualization, dynamic provision of computing resources as a utility.

Table 2. A Comparison between Peer-to-Peer Computing, Cluster Computing, Grid Computing and Cloud Computing

Criteria	Peer-to-Peer Computing	Cluster Computing	Grid Computing	Cloud Computing
Virtualization	Limited	Limited	Half	Essential
Ownership	Shared Ownership	Single Ownership	Multiple Ownership	Single Ownership
Standards	No Standard	Virtual Interface Architecture	Some Open Grid Forum	Web Services (SOAP and REST)
Operating System	Windows or Mac OS or Linux	Windows or Linux	Any standard (dominated by UNIX)	A hypervisor runs multiple OS
Resource Management	Peer-to-peer	Centralized	Distributed	Centralized and Distributed
Application drivers	Content and file management and Instant messaging and games	Business and data centers and enterprise computing	High throughput scientific applications	Dynamically provisioned web applications
Capacity	Capacity increase automatically with popularity	Stable and guarantee capacity	Different, but large capacity	On demand dynamically provisioned capacity
Failure management	It handle failure by pro- viding special nodes, called relays, that store any updates temporarily until the destination reappears on the network	Limited (often failed task restarted)	Limited (usually failed jobs restarted)	Failover, content replication, migration of VMs supported
SLA	SLA Based	Limited	SLA Based	Essential
Security	Low	Very low- but typically high	High	Low

4 POST CLOUD COMPUTING PARADIGMS

Ubiquitous/Pervasive computing [34] is regarded as a leading technological path of innovation. Ubiquitous computing is the rising trend of embedding computational capabilities (mainly in the form of microprocessors) into everyday objects to allow them to communicate properly and accomplish beneficial tasks in manner that reduces the need for end user to communicate with computers. Pervasive computing resources are connected to the network and constantly available. In contrast to desktop computing, pervasive computing can occur with any device, at any time, in any place and in any data format through any network and perform tasks from one computer to another. Ubiquitous computing typically includes wireless communication and networking technologies, mobile devices, embedded systems, wearable computers, radio frequency ID (RFID) tags, middleware and software agents. Internet capabilities, voice recognition and artificial intelligence (AI) are also often addressed. Pervasive computing applications were designed for user use and to help people perform their jobs. An example of pervasive computing is an Apple Watch that alerts the user to a phone call and enables the call to be finished through the watch.

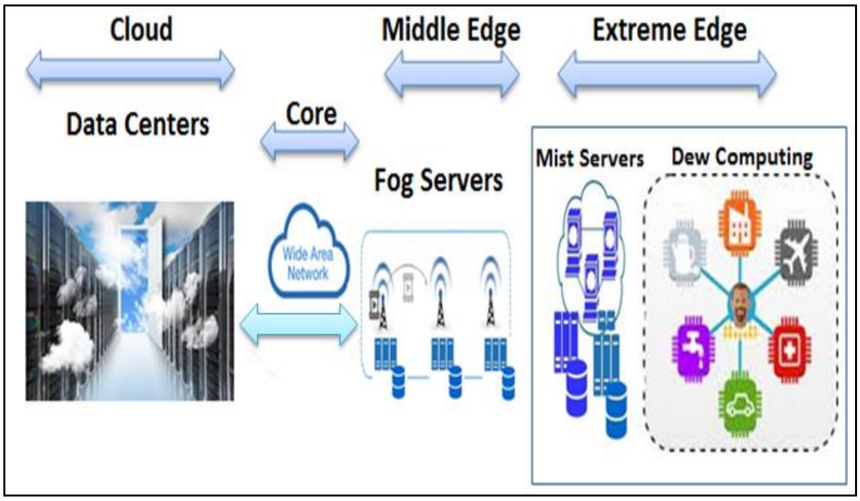

Fig. 5. Edge networks levels

With the quick technological growth of pervasive intelligent devices and ubiquitous networks, besides the new associated networked applications, the cloud computing model with its centralized processing and storage model is difficult to adapt and extend to different technologies and application scenarios. For this reason the industrial and academic communities have been proposing new network computing paradigms for the post-cloud computing age since 2011 [35]. Meanwhile, fog computing [36], Edge Computing [37], [38], mist computing, dew computing [7], and other post-cloud computing paradigms have been suggested and evolved. Although the post-cloud computing models according to their objectives, technologies, and application areas, their basic idea is the same [39]. That is, they try to physically or logically build cloud infrastructures closer to end users and their devices and then use the computing and storage resources in these local infrastructures to quickly finish the needed computation or storage operations of the end users, thus speeding up processing and response time and enhancing user experience.

The delay between IoT devices and cloud, the load on cloud and communication network across sensed data transmission developed the concept of computing on the edge of the network [6]. The levels of edge networks (dew, mist and fog) and the way they are connected to the cloud by the internet are shown in Fig. 5. Extreme Edge is called dew. They are actually the end IoT devices. On the extreme edge, the delay is almost insignificant since there is no communication necessary to make any decision with any other node. Dew computing is utilized when the application needs real time decision making without delay. This operates well if the computing power demand and previously sensed data is too small. In these situations, dew computing is highly efficient and saves energy, bandwidth etc.

A server is set up very close to the IoT device in Mist computing, typically in the same building and interconnected through a wireless/wired network. It is sufficient if the power of processing and the previously requirement of the sensed data exceeds the dew level capability. In mist, the delay is also too small. The communication distance ranges between a few-meters and few-hundred meters.

Middle edge or fog links extreme edge to the cloud through core network (WAN) [40]. Fog may be as distant as a few kilometers from the devices of IoT. The Fog server may be accessible with the Base Station (BS) of an Internet Service Provider (ISP) or in the network separately. The need for processing power and storage in the fog system is higher than in mist. There may be many mist associated with one fog. The fog distance from the IoT devices exceeds dew/mist, however it is recommended due to its higher processing power and minimal delay contrary to those of mist and dew. The core is the WAN, consisted of all the classical network components such as routers, bridges etc. If Cloud is considered at the top, then it generates networks hierarchy with dew, mist and fog on the lower levels.

4.1 Fog Computing

Fog computing [41] was initially developed by Cisco in 2012 to overcome the problems of IoT applications in traditional Cloud computing. IoT devices/sensors are widely distributed at the edge of the network together with real time and latency sensitive application needs. Cloud datacenters usually fail to manage processing and storage because they are geographically centralized. Thus, congested network, high service delivery latency, low Quality of Service (QoS) are encountered so Fog computing has been launched as a new computing technology. Fog computing is a network architecture that spans from the point of data creation to data storage location, whether it is cloud or local data center. Fog computing serves as an intermediate layer among Cloud datacenters and IoT devices/sensors. Fog computing is the expansion of the cloud to the network edge. Fog computing enables decentralized computing through processing data at the fog node. Any computer capable of storing, computing, and connecting to the network can be used as a fog node. Fog computing also enables mobility support, location awareness, real-time interactions, interoperability and scalability [42]. A Fog computing system essentially composed of traditional networking equipments such as switches, routers, proxy servers and Base Stations (BS), etc. and may be positioned nearer to the proximity of IoT devices/sensors.

Fog computing has a significant benefit to smart cities, since many devices utilize real time data to handle different tasks. Fog computing is utilized also in autonomous vehicles as data processing needs to be done in real time. The advantages of fog computing are:

- Fog computing enables real time data analysis that allows IoT applications work faster.
- Businesses decrease storage and computational expenses by processing data at fog nodes. Moreover, confidential data will be secured since it is stored at the fog node.
- Fog computing are used to improve low latency networks among analytics endpoints and devices. Compared to cloud computing, using such net- works will lead to reduction of bandwidth requirements.
- Fog computing can process greater volumes of data compared to edge computing because it can manage requests in real time.

The disadvantage of fog computing is that:

- The fog computing relies on multiple links to transfer data from the physical asset chain to the digital layer that will be potential points of network failure.

4.2 Edge Computing

Edge computing [43] is used directly to process data on devices which have joined sensors or gateway devices which are near the sensors. Hence, edge computing can enable devices to process data without relying on the cloud or fog. By processing data closer to the edge, edge computing can enable devices to process data in near real time. Edge computing will minimize the overhead at the centralized cloud. Edge computing will be utilized in smart homes to do tasks like switching on the heater or lights in near real time. Also, edge computing automates organizational predictive maintenance by submitting instant warnings about possible equipment defects. The advantages of edge computing are:

- Edge computing may facilitate internal communication by cabling IoT devices with physical assets to capture and process critical data.
- After data processing, IoT devices will choose which data should be sent to the cloud for analysis and which data should be stored locally. In this way, sensitive data will be stored discretely at its source.
- In addition, devices using edge computing will deliver near real-time analytics that will help in improving performance and expanding up-time.

Disadvantages of edge computing are:

- Edge computing is less scalable contrasted with fog computing.
- Edge computing supports low levels of interoperability that could make IoT devices inconsistent with some cloud services and operating systems.
- Multiple tasks and operations managed by IoT devices and cloud cannot be extended to an IT team.
- Edge computing does not allow the pooling of resources.

4.3 Mist Computing

Mist computing is used at the network extreme edge that is composed of micro-controllers and sensors. Through operating at the extreme edge, mist computing will use the processing and communication capabilities available on the sensor to extract resources. Mist computing infrastructure is using micro controllers and microcomputers to send data to fog computing nodes and subsequently to the cloud. On the sensor itself, arbitrary computations can be performed and controlled using this network infrastructure. Mist computing will be extremely useful for IoT in public transportation as the devices cannot be stationary and can serve only a singular purpose. The advantages of the mist computing are:

- Mist computing allows local decision making with the aid of sensors and micro controllers.
- Mist computing preserves bandwidth and battery power because important data is allocated to the gateway, server, or router.
- Additionally, mist computing allows the usage of data access control techniques to guarantee data privacy at a local level.

The disadvantages of mist computing:

- Microcomputers and sensors used in the mist computing infrastructure can only be used for lightweight data processing and a limited range of tasks. It is possible to utilize these tools for specific applications.

4.4 DEW COMPUTING

Dew computing [45] relies on the idea of micro services that are offered by the devices of end user (e.g. robots, mobile-devices, smart-objects or laptops) without the support of centralized virtual resources. In the cloud computing model, the end-user devices / onsite devices are geographically dispersed simply run fully online applications using cloud services. In the Dew Computing approach, many properties and data are transferred to the onsite devices, completely understanding the power of cloud services and distributed devices. The Dew computing model may coexist with the cloud and the fog models: the on-site devices can communicate with central computing nodes whenever the scenario permits it and the Internet connection is available, although they are not depending on them. The primary objective of Dew Computing is to en- hance scalability so the processing tasks are relatively distributed across a wide range of devices, that are self-adaptive, heterogeneous and adhoc programmable. Therefore, highly distributed applications can be realized without the usage of central nodes [46].

The Dew computing is placed as the ground level for the paradigms of cloud and fog computing in the current computing hierarchy. Compared with fog computing, that serves innovative IoT applications which require predictable latency, real time and dynamic network reconfiguration, Dew computing pushes the outlines to low level services, computing applications and data far away from centralized virtual nodes to the end-users.

A comparison between cloud, edge, fog, mist and dew computing is shown in Table 3 [44].

Table 3. A Comparison between dew, mist, edge, fog and cloud computing

Criteria	Dew Computing	Mist Computing	Edge Computing	Fog Computing	Cloud Computing
Service location	At edge network	At edge network	In edge network	Within the Internet	Within the Internet
Distance of (number loops)	No loop	Single loop	Single loop	Multiple loops	Multiple loops
Latency	Negligible	Very low	Low	High	Very high
Jitter	Negligible	Very low	Low	High	Very high
Location awareness	Yes	Yes	Yes	No	No
Geo-distribution	Highly distributed	Highly distributed	Distributed	Semi centralized	Centralized
Mobility sup-port	Highly supported	Highly supported	Semi supported	Limited	Very limited
Target users	Purely mobile users	Semi mobile users	Semi mobile users	General internet users	General internet users
Service scope	Purely limited	Purely limited	Semi limited	Semi global	Semi global
Hardware	Very limited capabilities	Limited capabilities	Limited capabilities	Scalable capabilities	Scalable capabilities
User experience	Highly satisfactory	Very good	Good	Normal	Very Normal
Internet dependency	Not essential	Not essential	Every access time	Every access time	Every access time
Client-Server connectivity	No	Yes	Yes	Yes	Yes
Synchronization feature	Always essential	Essential	Not essential	Not essential	Not essential
Delay tolerant	Yes	No	No	No	No
Computational offloading	Very High	High	High	Less	Very Less
Deployment scenario	PC, laptop, smart phone	microcontrollers and sensors	Router, gateway	Small sized (SME) to Medium Enterprises	Large enterprises

5 JUNGLE COMPUTING

In spite of the fact that there is an obvious requirement for programming solutions which enable scientists to acquire high-performance and distributed computing both effectively and transparently, real solutions are still missing [47].. Furthermore, there is already a revolutionary change in the high-performance and distributed computing world. Current clusters, grids and cloud networks are increasingly fitted multi-core technologies (for example, graphics processing units or GPUs) [48]). While these tools often provide order-of- magnitude speed enhancement, they allow computing systems more heterogeneous and hierarchical, and significantly more complex to and use and program. More complications emerge throughout daily practice causing the increasing need for computing power, and due to significant problems involving software heterogeneity, data distribution and ad-hoc hardware availability, scientists are often pushed to use several clusters, grids, clouds, and other systems at the same time even for single applications.

Jungle Computing [17] arises from the abundance of available distributed resources. A Jungle Computing System comprises of all available computing resources to end-users, including clusters, clouds, grids, desktop grids, supercomputers, as well as stand-alone machines and mobile devices. Jungle Computing Systems are used for many purposes. First of all, an application will need more computational power than any program that is accessed by the user. Second, various elements of an application can have different computational specifications, with no single system fulfilling all specifications.

All resources in a Jungle Computing System is somehow or another equivalent, all composing of a certain amount of storage, processing power and memory. End users view these resources as a compute resource for running their application on. This resource is either existed in a remote cloud or located down the hall in a cluster; it is of no concern to an end-user, as long as long as its application is operating efficiently. A Jungle Computing System is extremely heterogeneous in spite of this resource similarity. Resources are different in fundamental properties including processor architecture, performance and quantity of memory. Installed applications such as compilers and libraries will also vary as there is no unified administration of these different systems.

For instance, when a standalone machine is often available, a grid resource must be reserved, although a cloud needs a credit card to use it. Furthermore, due to the use of various interfaces, the middleware utilized to access a resource is very different. The heterogeneity of Jungle Computing Systems makes it difficult to execute applications on multiple resources. For every utilized resource, the application may need to be re-compiled or indeed partially re-written, to handle the adjustments in the available software and hardware. In addition, a different middleware interface can be needed for every resource, having different middleware client software. Another feature that inhibits the use of Jungle Computing Systems is the absence of resources connectivity. The Ibis high-performance distributed programming framework is an example of software platforms designed to assist Jungle computing. It has become exceedingly difficult to write applications for such Jungle Computing Systems, particularly with the introduction of multi-core hardware technologies.

The objective of the Ibis platform is to significantly simplify Jungle Computing applications programming and deployment. Ibis implements solutions to many of Jungle Computing's basic problems into a single modular programming and deployment system, written entirely in Java. Ibis provides Jungle Computing with an effective and straightforward solution, there is an urgent need for Ibis to become a feasible programming system for daily scientific practice. One of the main questions to be answered is if a collection of fundamental building blocks can be described that can characterize any application for Jungle Computing. These components can be used to describe both generic models of programming (e.g., MapReduce, SPMD, divide and conquer and pipelining), and domain-specific models (e.g., the Jorus model). Another question is whether all of these models indeed actually deliver effective execution on different Jungle Computing Systems. In this regard, the question is whether generic computational patterns will be replicated to describe a variety of domain-specific programming models. The availability of such generic patterns would significantly increase the emergence of new programming models for undiscovered scientific domains. Jungle Computing Challenges:

- There are often several kernels with the same features but aimed at various platforms (referred to as equi-kernels). All of these kernels are beneficial, e.g. due to various scalability features or availability of ad hoc hardware. The challenge is to transparently integrate (multiple) domain-specific kernels with Jungle Computing programming models and applications. More approaches should be explored, including those that take into consideration the major advantages of coordinating several sub- sequent kernels, and scheduling these as a single kernel.
- Kernels mapping to resources is a dynamic issue. It is due to the possibility of adding or removing resources and the computational requirements of kernels that vary over time. In addition, the mapping can take into consideration optimization under several, probably overlapping, goals (e.g., speed, energy use, financial costs and productivity). The problem is to what degree the transparent and dynamic migration of compute kernels in Jungle Computing Systems can be enabled with run- time support.

6 Conclusion

In this paper, we discussed the past, present and future of scalable computing technologies. First, we classify computing technologies into high performance computing, distributed computing and Jungle computing. High performance computing is effective in all compute-intensive research fields since it is appropriate to tasks which are either numerically, computationally or data intensive. Exascale computing and quantum computing are the most recent and hot topics to efficiently achieve High performance computing. Since both supercomputing and quantum computing have their advantages and disadvantages so it is recommended to implement a hybrid system that uses both technologies.

Supercomputers have a very high cost so distributed computing systems that provide high performance along with versatility and cost efficiencies are developed. In the past decade the cloud computing technology is acquiring a great amount of prominence as a distributed computing paradigm. For sure, it is impacting the improvement of information-technology and modifying the manner that vendors, providers and end-users consider computation.

The cloud computing has a centralized processing and storage model that is difficult to adapt post cloud computing technologies. Fog computing, edge computing, mist computing and dew computing have their own advantages and disadvantages. Understanding and utilizing those technologies correctly with cloud computing can help in designing secure, reliable, and highly operational IoT solutions.

Acknowledgment

The authors send their acknowledge to STDF for their financial grant of the cloud computing center of excellence (number 5220).

References

[1] D. Sood, H. Kour, S. Kumar, "Survey of computing technologies: Distributed, utility, cluster, grid and cloud computing", Journal of Network Communications and Emerging Technologies (JNCET), vol. 6, pp. 99–102, 2016.
[2] M. Obali, A. E. Topcu, "Comparison of cluster, grid and cloud computing using three different approaches", 23nd Signal Processing and Communications Applications Conference (SIU), pp. 192–195, 2015.
[3] R. Guharoy, S. Sur, S. Rakshit, S. Kumar, A. Ahmed, S. Chak- borty, S. Dutta, M. Srivastava, "A theoretical and detail approach on grid computing a review on grid computing applications", 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 2017.
[4] B. K. Rani, B. P. Rani, A. V. Babu, "Cloud computing and inter clouds types, topologies and research issues", Procedia Computer Science, vol. 50, pp. 24–29, 2015.
[5] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, A. Zanni, "A survey on fog computing for the internet of things, Pervasive and Mobile Computing", vol. 52, pp. 71–99, 2019.
[6] M. Satyanarayanan, "The emergence of edge computing", Computer, vol. 50, pp. 30–39, 2017.
[7] Y. Wang, "Definition and categorization of dew computing", Open Journal of Cloud Computing (OJCC), vol. 3, pp. 1–7, 2016.
[8] M. Tripathy, C. R. Tripathy, "A comparative analysis of some high performance computing technologies", International journal of advanced computer technology, vol. 3, pp. 1149–1156, 2014.
[9] M. N. O. Sadiku, S. M. Musa, O. M. Musa, "High-performance computing: A primer", International Journal of Advanced Research in Science, Engineering and Technology, vol. 4, pp. 4661– 4663, 2017.
[10] E. Abraham, C. Bekas, I. Brandic, "Preparing hpc applications for exascale: challenges and recommendations", 18th Int. Conf on Network-Based Information Systems, pp. 401–406, 2015.
[11] X. LIAO, K. LU, C. YANG, J. LI, Y. YUAN, M. LAI, L. HUANG, P. LU, J. FANG, J. REN, J. SHEN, "Moving from exascale to zettascale computing: challenges and techniques", Front Inform Technol Electron Eng, vol. 10, pp. 236–1244, 2018.
[12] T. A. Shaikh, R. Ali, "Quantum computing in big data analytics: A survey", 2016 IEEE International Conference on Computer and Information Technology, pp. 112–115, 2016.
[13] X. Hu, M. Niemier, "Cross-layer efforts for energy efficient computing: towards peta operations per second per watt", Front In- form Technol Electron Eng, vol. 19, pp. 1209–1223, 2018.
[14] Performance development, https://www.top500.org/ statistics/perfdevel (2019).
[15] J. Zhai, W. Chen, "A vision of post-exascale programming", Front Inform Technol Electron Eng, vol. 19, pp.1261–1266, 2018.
[16] Z. Xu, X. Chi, N. Xiao, "High-performance computing environment: a review of twenty years of experiments in china", International journal of advanced computer technology, vol. 3, pp. 36–48, 2016.

[17] B. Kahanwal, T. P. Singh, "The distributed computing paradigms: P2p, grid, cluster, cloud, and jungle", International Journal of Latest Research in Science and Technology, vol. 1, pp.183–187, 2015.
[18] H. Wang, H. Takizawa, H. Kobayashi, "A dependable peer-to- peer computing platform", Future Generation Computer Systems, vol. 23, pp. 939–955, 2007.
[19] D. Castella, F. Solsona, F. Gine, "Discop: A p2p framework for managing and searching computing markets", Journal of Grid Computing, vol. 13, pp. 115–137, 2015.
[20] C. P. Miguel, J. M. Alonso, A. Mendiburu, "High throughput computing over peer-to-peer networks", Future Generation Computer Systems, vol. 29, pp. 352–360, 2013.
[21] P. Lopez, E. Baydal, "Teaching high-performance service in a cluster computing course", Journal of Parallel and Distributed Computing, vol. 117, pp.138–147, 2018.
[22] M. Zakarya, L. Gillam, "Energy efficient computing, clusters, grids and clouds: A taxonomy and survey", Sustainable Computing: Informatics and Systems, vol. 14, pp. 13–33, 2017.
[23] H. Singh, G. Singh, "Task scheduling in cluster computing environment", International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 316–321, 2015.
[24] M. B. Qureshi, M. M. Dehnavi, N. M. Allah, M. S. Qureshi, H. Hussain, I. Rentifis, N. Tziritas, T. Loukopoulos, S. U. Khan, C. Z. Xu, A. Y. Zomaya, "Survey on grid resource allocation mechanisms", J Grid Computing, vol. 12, pp. 399–441, 2014.
[25] S. N. Pardeshi, C. Patil, S. Dhumale, "Grid computing architecture and benefits", International Journal of Scientific and Research Publications, vol. 3, pp. 1–4, 2014.
[26] J. Yu, R. Buyya, "A taxonomy of workflow management systems for grid computing", Journal of Grid Computing, vol. 3, pp. 171– 200, 2006.
[27] L. Li, Y. Wang, Y. Yang, Z. Tian, "Utility-based computing model for grid", Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG 2005), 2005.
[28] Q. Chen, Q. N. Deng, "Cloud computing and its key techniques", Jr. of Computer Applications, vol. 4, pp. 25–62, 2009.
[29] N. Sadashiv, S. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison", Proceeding of the 6th International Conference on Computer Science and Education (ICCSE), pp. 477–482, 2011.
[30] B. Rajkumar, Y. C. Shin, V. Srikumar, B. James, B. Ivona, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, pp. 599–616, 2009.
[31] T. Grance, P. Mell, The nist definition of cloud computing, National Institute of Standards and Technology (NIST).
[32] A. Lenk, A. Klems, J. Nimis, S. Tai, T. Sandholm, "What's inside the cloud? An architectural map of the cloud landscape", IEEE, Washington, DC, USA.IEEE Computer Society, 2009.
[33] Z. Qi, C. Lu, B. Raouf, "Cloud computing: state-of-the-art and research challenges", Springer, pp. 7–18, 2010.
[34] K. Sakamura, N. Koshizuka, "Ubiquitous computing technologies for ubiquitous learning", IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'05), 2006.
[35] D. Sehrawat, N. S. Gill, "Emerging trends and future computing technologies: A vision for smart environment", International Journal of Advanced Research in Computer Science, vol. 9, pp. 839–842, 2018.