INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
http://dx.doi.org/10.1080/10798587.2017.1280995

Recent Advances in Mobile Grid and Cloud Computing

Sayed Chhattan Shah

Department of Information Communication Engineering, Hankuk University of Foreign Studies, Yongin, South Korea

ABSTRACT
Grid and cloud computing systems have been extensively used to solve large and complex problems in science and engineering fields. These systems include powerful computing resources that are connected through high-speed networks. Due to the recent advances in mobile computing and networking technologies, it has become feasible to integrate various mobile devices, such as robots, aerial vehicles, sensors, and smart phones, with grid and cloud computing systems. This integration enables the design and development of the next generation of applications by sharing of resources in mobile environments and introduces several challenges due to a dynamic and unpredictable network. This paper discusses applications, research challenges involved in the design and development of mobile grid and cloud computing systems, and recent advances in the field.

KEYWORDS
Cloud Robotics; Sensor Cloud; Mobile Distributed Systems; Mobile Edge Cloud

1. Introduction

A distributed system consists of a collection of autonomous computers that are connected through a network and distribution middleware, which enables computers to coordinate their activities and share the resources of the system so users can perceive the system as a single, integrated computing facility (Wolfgang, 1997).

One of the best examples of a distributed system is an automated teller machine, where a user is provided with a simple and easy-to-use interface to perform numerous transactions. From the user's perspective, there is a single system, but in the background, hundreds of different computing devices are connected through various networks to provide a range of financial services.

The distributed systems are divided into three main categories; cluster, grid and cloud computing systems. In cluster, distributed computing devices are connected through a high-speed local area network, whereas in grid, geographically distributed resources are connected through a wide area network to solve large and complex problems.

A layered architecture of cluster and grid computing systems is presented in Figures 1 and 2, respectively. The resource layer includes computing devices that are connected through networking and communication technologies. The middleware layer provides resource and task management services, which include the resource monitoring and discovery service, fault management service, resource allocation service and task migration service. Moreover, middleware hides all of the complexities and provides a single system image to both the user and the applications running on the system. Resource and task management in clusters are usually based on a centralized architecture, whereas grid relies on distributed architectures for resource and task management.

Cloud computing has evolved from cluster and grid computing and is an integration of various concepts and technologies, such as hardware virtualization, utility computing, autonomic computing, pervasive computing and service-oriented architecture. In cloud computing, everything from computing power to communication infrastructure and applications is delivered as a service over a network (Mell & Grance, 2011). Cloud computing technologies and a service model are given in Figure 3.

The basic idea is to provide computing and communication resources on demand for a fee, just like the electrical power grid. Users should be provided with an easy-to-use interface to access vast amounts of cloud resources without being concerned with details such as how the power is generated or how various resources are connected.

Grid and cloud computing systems have been extensively used to solve large and complex problems in science and engineering fields, such as drug design, earthquake simulation, and climate modeling. These systems include powerful computing resources that are connected through high-speed networks.

Due to the recent advances in mobile computing and networking technologies, it has become feasible to integrate various mobile devices, such as robots, aerial vehicles, sensors, and smart phones, with grid and cloud computing systems. The approaches for integrating mobile devices with grid and cloud computing systems are divided into two main categories; mobile grid and cloud computing and mobile ad hoc grid and cloud computing. Both categories are further divided into two sub-categories; data grid and cloud and computational grid and cloud. The taxonomy of distributed systems is given in Figure 4. This paper discusses the applications, research challenges involved in the design and development of mobile grid and cloud computing systems, and recent advances in the field.

The paper is classified into the following sections: Section 2 describes concepts, applications and types of mobile grid and cloud computing systems. Section 3 focuses on mobile ad hoc grid and cloud computing systems. Challenges involved in the design and development of mobile grid and cloud computing systems are discussed in Section 4. Section 5 describes

CONTACT Sayed Chhattan Shah shah@hufs.ac.kr

selected research projects and Section 6 discusses state-of-the-art mobile grid and cloud computing systems, architectures and schemes. Section 7 presents conclusion.

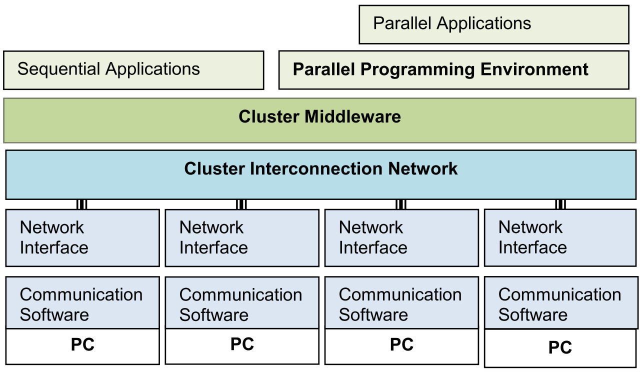

Figure 1. Cluster Architecture.

2. Mobile Grid and Cloud Computing

2.1. Mobile Cloud Computing

In mobile cloud computing, mobile devices are integrated with a cloud computing system through an infrastructure-based communication network such as a cellular network. The computationally intensive tasks are offloaded to a cloud for execution. The architecture of a mobile cloud system is given in Figure 5.

2.2. Mobile Grid and Cluster Computing

In mobile grid computing, mobile devices are connected to a grid computing system through an infrastructure-based communication network, such as a cellular network. The computationally intensive tasks are sent to a grid computing system that, after processing, sends the results back to the mobile device.

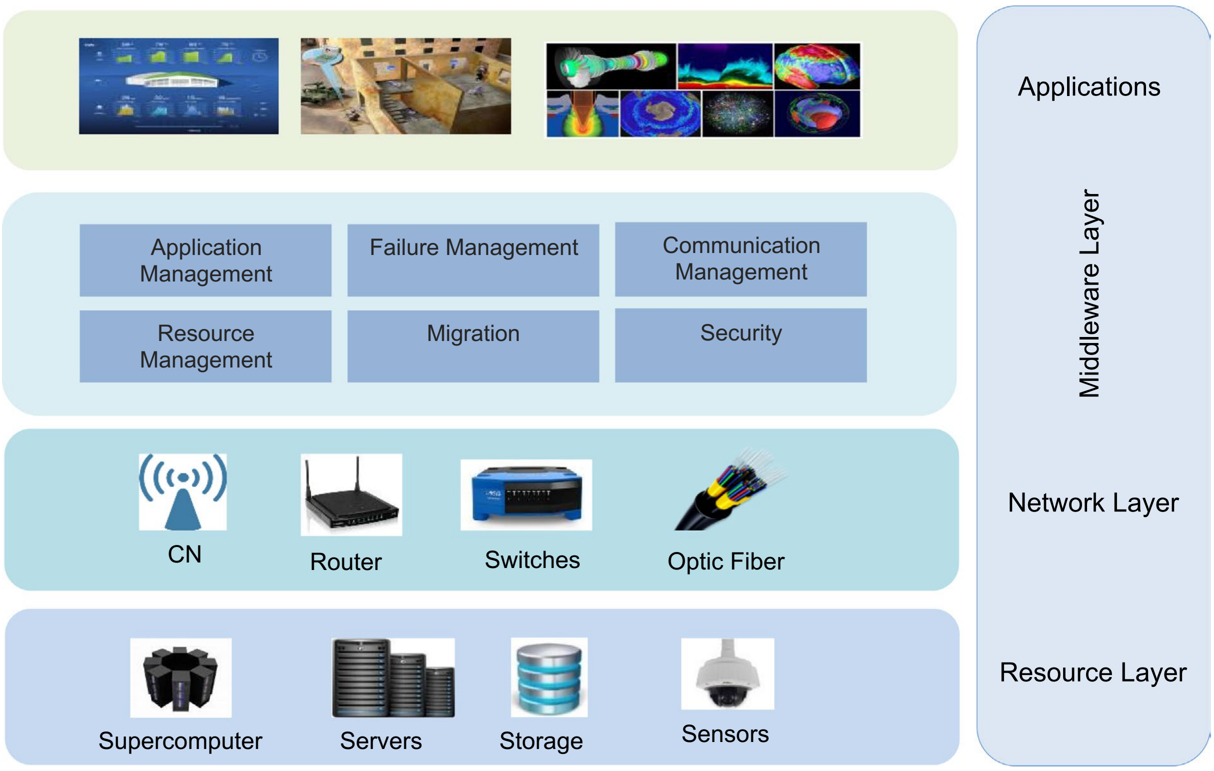

Figure 2. Grid Architecture.

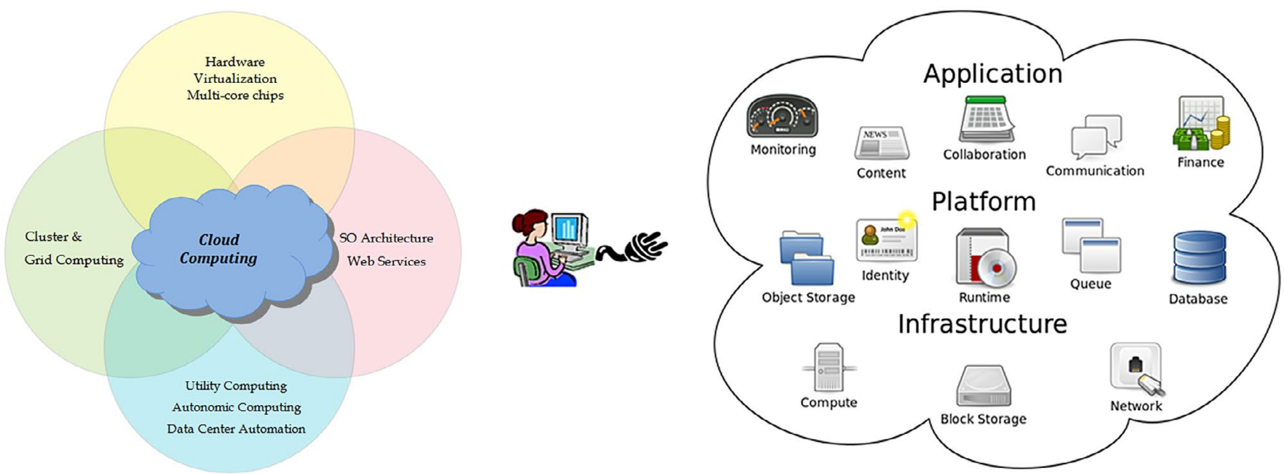

Figure 3. Cloud Technologies and Service Model.

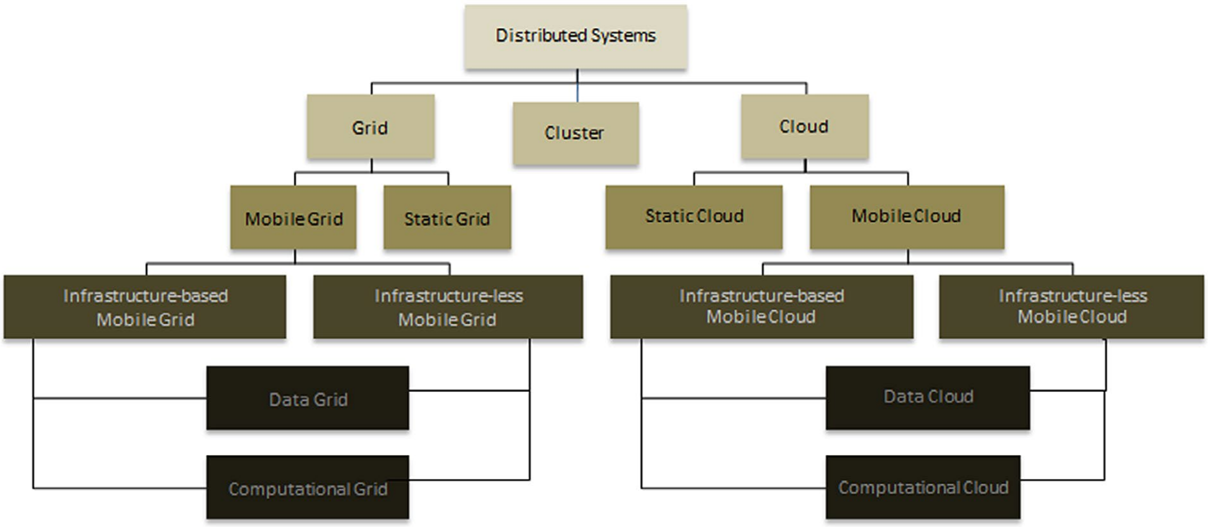

Figure 4. Taxonomy of Distributed Systems.

Figure 5. Mobile Cloud System Architecture.

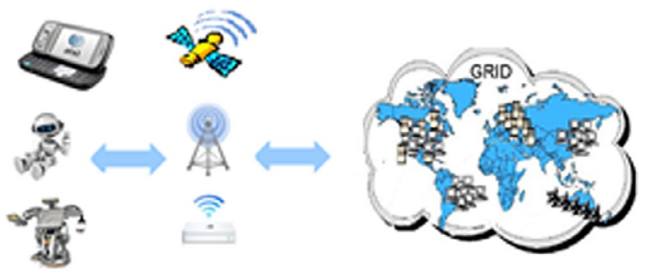

Figure 6. Mobile Grid System Architecture.

In mobile cluster computing, mobile devices are connected to a cluster through an infrastructure-based network. The architecture of a mobile grid system is given in Figure 6.

2.3. Enabling Factors

The idea of integrating mobile devices with cluster and grid computing systems was introduced in the early 1990s, but did not become popular due to the limited processing power of mobile devices and the low speed of wireless networks.

The continuous increase in the processing power of mobile devices, the improved data transfer rates of wireless networks, and the growing popularity of cloud computing systems and mobile devices have made it feasible and valuable to integrate mobile devices with traditional distributed systems. The currently available mobile devices, such as the iPhone 6, have more processing power than did the supercomputer used in the early 1990s (Nick, 2014; White, 2013), and the current generation of wireless networks, such as 4G, provide a data rate of 100 megabits per second, which is sufficient for streaming a video. Several companies are already testing 5G networks that aim to provide a data rate of 10 gigabits per second. Figure 7 illustrates the CPU speed of an iPhone 4 compared to that of the Cray-2 supercomputer and the performance of cellular network technologies.

2.4. Benefits

Cloud computing systems provide a vast amount of storage space and processing power on demand. The integration of mobile devices with cloud computing systems enables mobile devices to access vast amounts of processing power and storage space. This makes it possible to execute data and computationally intensive applications, such as image and video processing, on mobile devices. Data storage and execution of such applications on the cloud also improve reliability and extend the battery life of mobile devices.

2.5. Cloud Robotics

Cloud robotics is another interesting topic in the area of mobile cloud computing. In cloud robotics, robots are integrated with a cloud computing system to access vast amounts of processing power and data storage (Bonaccorsi, Fiorini, Cavallo, Esposito, & Dario, 2015; Mohanarajah, Hunziker, D'Andrea, & Waibel, 2015; Wan et al., 2016).This integration allows robots to offload heavy tasks, such as image processing and voice recognition, to the cloud. A block diagram of cloud robotics is given in Figure 8.

In addition to the advantages discussed for mobile cloud computing, the integration of robots with the cloud provides several other advantages.

1. The execution of computationally intensive tasks on the cloud would result in cheaper, lighter and easy-to-maintain hardware, a longer battery life, minimal software updates, and hassle-free and invisible CPU hardware upgrades (Erico, 2011).
2. Shared object library: Assume a scenario in which a robot deployed in an urban environment encounters an unknown object and is thus unable to manipulate it. To address this issue, one option is for the robot to maintain information about the commonly used

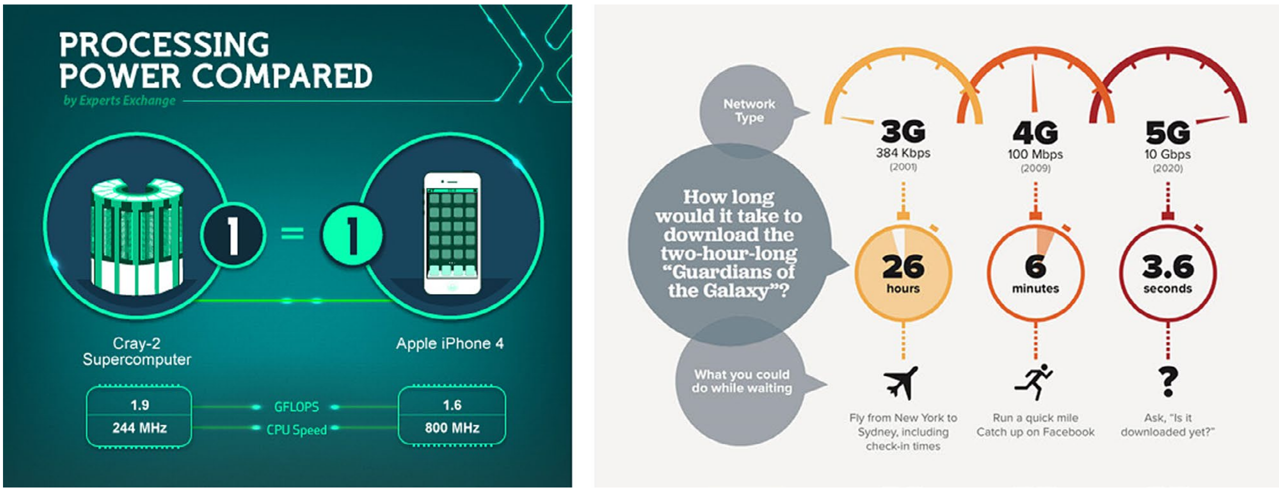

Figure 7. Processing Power of iPhone 4 and Cray-2 Supercomputer and Performance of Wireless Cellular Networks (Shankland, 2015).

Figure 8. Cloud Robotics.

objects, which would require much storage space and also processing power. The second option is to maintain the information of commonly used objects on the cloud. When a robot encounters an unknown object, it can take a picture of the object and send it to the cloud, which in turn will return the required information, such as the object name, 3D model, mass, materials, friction properties, and usage instructions, that are required to operate the object. The object library on the cloud can be shared by several robots and can be updated regularly by experts.

3. Similar to an object library, a library of numerous skills, navigation algorithms, and maps can also be maintained on the cloud. When a robot is deployed in an unknown environment and is unable to navigate, it can directly download the required map and navigation algorithm to navigate in the environment. Moreover, the library may include several navigation algorithms developed by numerous experts. For example, when a robot has limited battery power, it may choose to download an energy efficient algorithm, but if energy is not a concern, a computationally intensive algorithm with better performance can be used.

2.6. Sensor Grid and Cloud

In sensor grid and cloud, sensor nodes are integrated with grid and cloud computing systems through a gateway node. Sensor nodes are used to collect data, which are then sent to the cloud for further processing. Sensor cloud architecture is presented in Figure 9. Sensor cloud has several applications in numerous areas, including cyber physical systems such as the smart home, smart grid and smart city (iCore, 2014). For example, in a smart aging project, several bio and environmental sensors, such as a camera, noise detector, temperature monitor, heart rate and blood pressure monitors, are deployed to observe the home environment and physiological health of an individual. The data collected by the sensors are sent to an application on the cloud, where numerous algorithms for emotion and sentiment detection, activity recognition, and situation detection are applied to provide healthcare and emergency services and manage resources at home.

3. Mobile Ad Hoc Computational Grid and Cloud

The mobile grid and cloud computing systems are restricted to infrastructure-based communication systems, such as cellular networks, and thus cannot be used in mobile ad hoc environments. Moreover, mobile grid and cloud computing systems face issues such as high transmission energy consumption and communication latency.

A mobile ad hoc computational grid is a distributed computing infrastructure that enables mobile nodes to share computing resources without a pre-existing network infrastructure (Shah, 2015), whereas a mobile ad hoc computational cloud is a distributed computing infrastructure in which multiple mobile devices, interconnected through a mobile ad hoc network, are combined to create a virtual supercomputing node.

The system architecture of a mobile ad hoc computational cloud is given in Figure 10.

A mobile ad hoc computational grid is a combination of a computational grid and mobile ad hoc network. A computational grid is a software infrastructure that allows distributed computing devices to share computing resources to solve computationally intensive problems. A mobile ad hoc network is a wireless network of mobile devices that communicate with each other without any pre-existing network infrastructure.

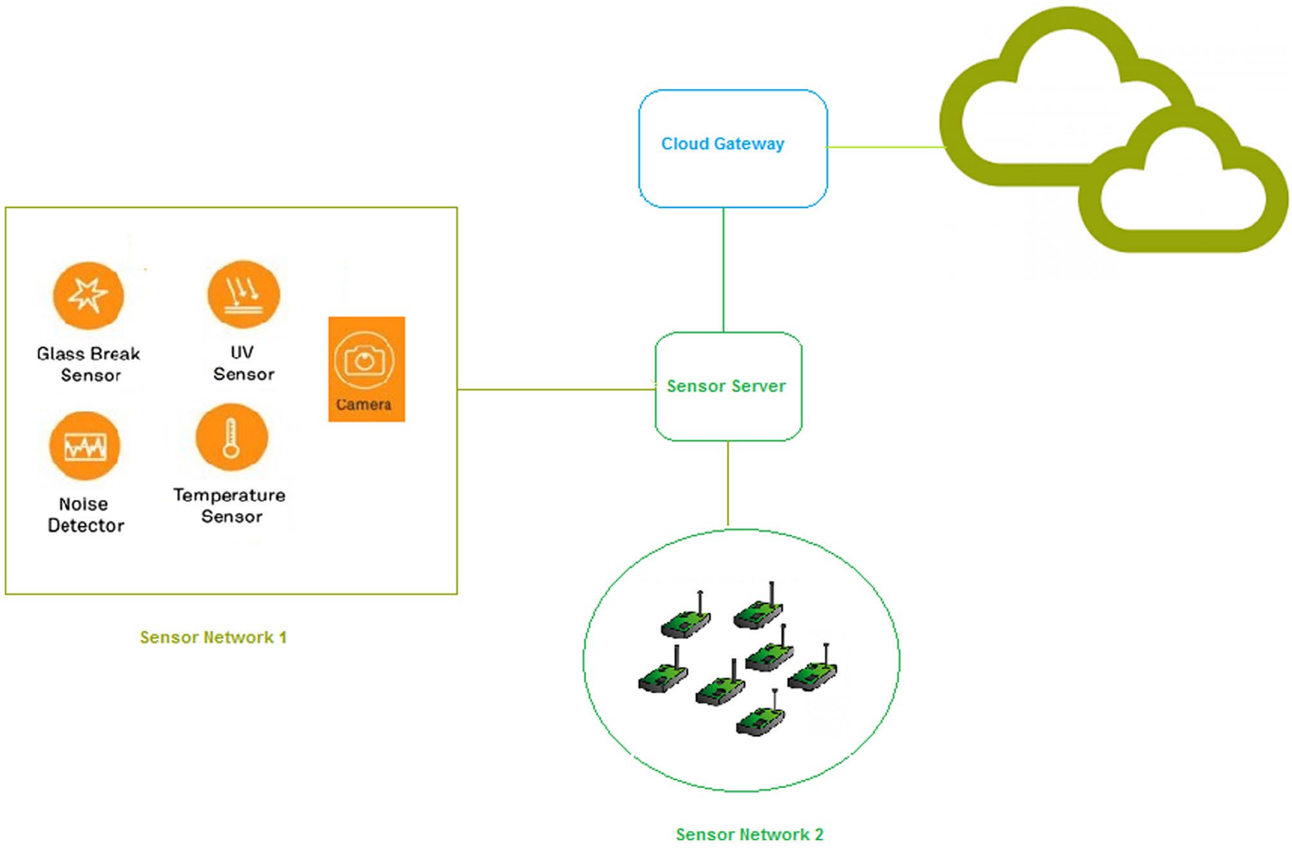

Figure 9. Sensor Cloud Architecture.

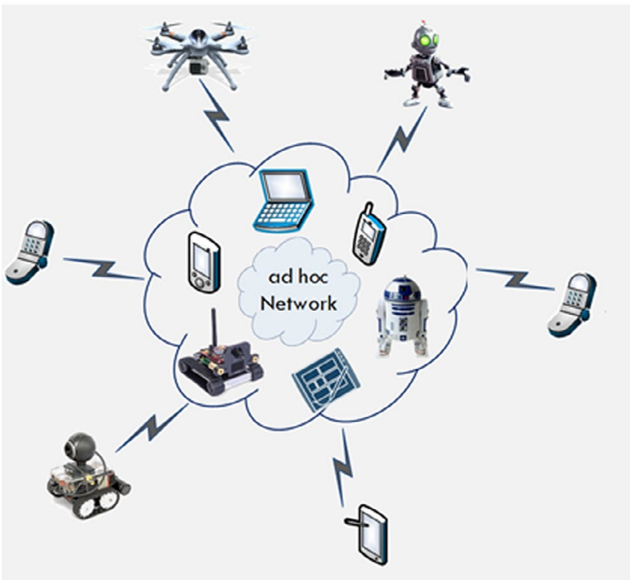

Figure 10. A Diagram of a Mobile Ad hoc Computational Cloud, in which Multiple Mobile Devices, Interconnected through a Mobile Ad hoc Network, are Presented *as One Powerful, Unified Computing Resource*.

The architecture of a mobile ad hoc computational grid is presented in Figure 11. Mobile nodes communicate with each other through a mobile ad hoc network, which provides several communication services, including transport, routing, and medium access. The middleware layer, i.e., the computational grid, provides the resource management service, which includes resource monitoring, resource discovery, and task management services, fault management service, mobility management service, communication service, and task migration service. Moreover, the middleware layer hides all of the complexities and provides a single system image to the user and applications running on the system.

3.1. Applications

A mobile ad hoc computational cloud would enable applications in numerous areas, such as security, disaster management, robotics, and smart environments. This section describes three key applications of the mobile ad hoc cloud.

3.1.1. Mobile Automated Intelligent Video Surveillance System

The success of a military or disaster relief operation depends on several factors, including an understanding of the physical environment and real-time detection and tracking of mobile and stationary targets. To understand the environment, various mobile robots and micro drones equipped with audio, video and environmental sensors are deployed to collect data, which is then processed to construct a three-dimensional map of the environment in real time. The collected data are also used to detect and track mobile and stationary targets, which involve sophisticated image and video processing algorithms (Garriss et al., 2008; Rellermeyer, Riva, & Alonso, 2008; Satyanarayanan, Bahl, Caceres, & Davies, 2009).

To perform these tasks, a vast amount of computing and storage resources are required. To address the issue, there are two options: 1 - Send data to an application on the cloud through an infrastructure-based communication system, such as a cellular network. 2 - Create a virtual supercomputing node comprising mobile robots, micro drones, and soldiers' wearable devices.

The first option is not feasible, because in the battlefield or a disaster relief operation, infrastructure-based communication

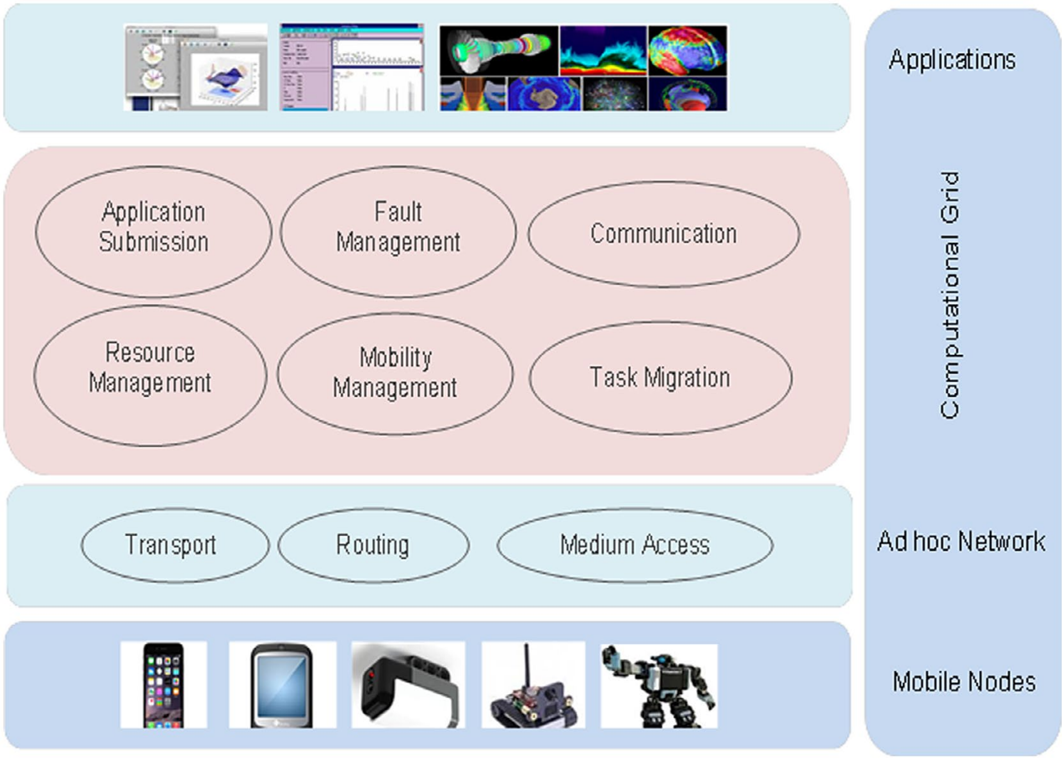

Figure 11. Mobile Ad hoc Computational Grid Architecture.

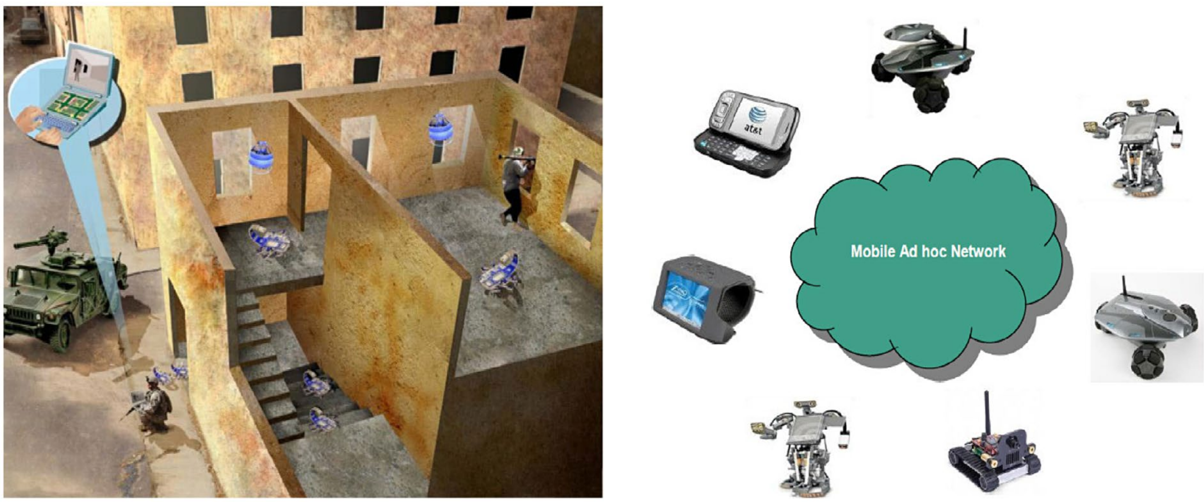

Figure 12. Automated Intelligent Video Surveillance System and Mobile Ad hoc Grid Architecture.

systems are usually not available, and even if they are available, their communication performance is much lower than that of short-range wireless communication technologies such as Wi-Fi Direct (Chao, Zhang, & Song, 2015). For example, compared to fourth-generation networks that provide a data rate of 100 megabits per second, a wireless local area network provides a data rate of up to 600 megabits per second. Figure 12 illustrates an automated intelligent video surveillance system and mobile ad hoc grid architecture.

3.1.2. Smart Home Environment

Several environmental, audio, video, and bio sensors, capable of acquiring vital signs, such as blood pressure, temperature, and electro cardiogram, are deployed to observe the home environment and physiological health of an individual. The data collected by sensors are sent to an application, where numerous algorithms for emotions and sentiments detection, activity recognition and situation detection are applied to provide healthcare- and emergency-related services and to manage resources at home. The execution of these computationally intensive real-time tasks requires computing capabilities that go beyond those of an individual sensing and processing devices. To realize this objective, the sensor nodes, wearable computing devices, smart phones and computing devices available at home can be combined to create an ad hoc computational cloud.

3.1.3. Mobile Gaming System

A group of friends are on a trip and would like to play a game in which human body tracking is used to animate the characters. To track the human body, mobile devices are equipped with video sensors, and we assume that one mobile device has a projector for display. This scenario requires tracking of human body

Figure 13. The Effect of Local Node Mobility. Tasks Executing on Nodes 1 and 2 Communicate Directly. During the Execution of a Task, Node 2 moves within the Network Coverage Area and therefore is not Directly Accessible to Node 1. Thus, a Task Executing on Node 1 Communicates with a Task Executing on Node 2 through Two Intermediate Nodes, i.e., Nodes 3 and 4. This Increases the Data Transfer Time and, thus, the Task Completion Time.

parts and other game-related tasks in real time. To realize this application, a group of mobile nodes can share their computing resources on demand to form a grid. From the grid, a sub-group of mobile devices can be assigned to each user to track his body movements; simultaneously, these devices can be used to execute other game-related tasks and communicate actions.

4. Research Challenges

Compared to traditional parallel and distributed computing systems, such as grid and cloud, mobile grid and cloud computing systems introduce several challenges, due to global and local node mobility, high latency, limited power, and a dynamic network environment. This section discusses the key research challenges.

4.1. Global and Local Node Mobility

Global node mobility refers to the movement of nodes across the coverage area, whereas local node mobility refers to the movement of nodes within the coverage area. Global node mobility results in the failure of one or several tasks, whereas local node mobility may increase data transfer cost and thus task completion time. The effect of local node mobility is described in Figure 13. A task migration or reallocation strategy can be used to manage task failure; however, this will introduce delay and increase the task completion time.

Global node mobility also introduces the problem of outdated information managed by resource discovery and monitoring system. To address this problem, the continuous monitoring of resources is required, but this will increase communication cost. An alternate approach is to enquire the status of resources during decision-making, but this will introduce delay.

To address the problems, an effective and robust resource allocation scheme has been proposed by Shah, Nizamani, Sajjad, and Park (2012). The scheme uses the history of a user's mobility pattern to select nodes that will remain connected for a long period of time. Because this scheme relies on the user's mobility history, it cannot be used in situations where mobility history is not available or the user has random and irregular mobility patterns.

4.2. Dynamic Communication Environment

The communication system of a mobile ad hoc grid and cloud is unpredictable, dynamic and limited in performance due to the low bandwidth, local and global node mobility, and lack of a preexisting network infrastructure. The connection quality at different network sections fluctuates over time. Even different nodes may experience different connection quality at the same time. Due to these reasons, data transfer cost is critical for application performance.

4.3. Power Management

Communication energy consumption depends on two key factors: Transmission power required to transmit data and communication cost induced by data transfers between tasks. In the literature, several mechanisms have been proposed to address the problem, but most of them are focused on the conservation of processing energy, while conserving communication energy remains an open problem, which becomes even more critical for data-oriented applications.

To reduce communication energy consumption, an energy efficient resource allocation scheme is developed by Shah and Park (2011). The main idea is to use a transmission power control mechanism and allocate dependent tasks to nodes that are accessible at minimum transmission power. However, further investigation is required to incorporate the transmission power control mechanism in resource discovery, resource monitoring, failure management, and task migration services.

4.4. Task Migration

Task migration primarily involves transferring both code and data across the nodes. Tasks are migrated for several reasons, such as to improve application performance and resource utilization, balance load, conserve energy, and manage task failure. Compared to traditional systems, the design of a migration service for mobile grid and cloud computing systems is difficult due to the dynamic network environment. The most common migration strategy is to estimate the migration cost and determine the task execution time with or without a migration. The estimation of migration cost depends on the estimation of data transfer time, which is not straightforward, particularly for data-oriented applications.

4.5. Computation Offloading

One of the key challenges in mobile grid and cloud computing systems is to devise an efficient task offloading strategy. In the literature, several offloading strategies have been proposed, which are classified based on numerous factors, such as;

Offloading objective: An application (Garriss et al., 2008; Satyanarayanan et al., 2009; Shilve et al., 2006) or part of an application (Giurgiu, Riva, Juric, Krivulev, & Alonso, 2009; Rellermeyer et al., 2008; Rellermeyer, Duller, & Alonso, 2009; Rellermeyer, Alonso, and Roscoe, 2007; Balan, Satyanarayanan, Park, & Okoshi, 2003; Chun, Ihm, Maniatis, Naik, & Patti, 2011; Cuervo et al., 2010; Dinh, Lee, Niyato, & Wang, 2013) is offloaded to a remote system for several reasons, such as to improve application performance, reduce energy consumption, or avoid task failure.

Offloading client: A mobile node that uses offloading could be a smart phone, mobile robot, autonomous aerial vehicle, or sensor.

Figure 14. Architectural Choices for Mobile Ad hoc Grid and Cloud Computing System.

Infrastructure for offloading: A mobile node may offload computation to a cluster (Busching, Schildt, & Wolf, 2012; Cheng, Wanchoo, & Marsic, 2000; Mohamed et al., 2005; Zheng, Buyya, & Bhattacharya, 1999), grid (Fox, Rui, & Isaac, 2008; Piotr & Poznan, 2005; Robinson, Frey, Clark, Reynolds, & Bedi, 2005), cloud computing system (Garriss et al., 2008; Giurgiu et al., 2009; Satyanarayanan et al., 2009; Shilve et al., 2006), or a powerful computer that is connected through a wide or local area network (Garriss et al., 2008).

Type of applications: Various application types have been studied, such as computationally intensive applications, data intensive applications, data parallel applications, and task parallel applications (Dinh et al., 2013).

Application partitioning mechanisms: To partition an application into tasks, two key approaches have been proposed; static and dynamic. The latter is more flexible and suitable for dynamic and unpredictable environments.

Offloading decision: A decision to offload a task could be made before runtime or during runtime.

To design an effective offloading strategy, several factors, such as network bandwidth, available energy, processing speed of the remote and host resource, task queue size, amount of data transfers between a task on a client node and a task to be allocated on a remote server node, should be considered.

4.6. Architecture

Every architecture have some pros and cons; e.g., a centralized architecture results in effective resource allocation decisions due to a network-wide view but suffers from scalability and single-point-of-failure problems. Meanwhile, a distributed architecture solves the scalability and single-point-of-failure problems, but results in poor resource allocation decisions due to lack of a network-wide view. Possible architectures for a mobile ad hoc grid and cloud computing system are given in Figure 14. There is a need to develop an energy-efficient, effective and robust architecture that should give maximum performance in a wide range of scenarios.

4.7. Parallel Programming Model

Programming models developed for infrastructure-based computing systems cannot be used in dynamic and ad hoc network environments. There is a need to design a programming model that provides lightweight migration and communication mechanisms and should support high mobility scenarios.

4.8. Transport and Routing Protocols

Numerous transport and routing protocols have been developed either to reduce data transfer times or transmission energy consumption. Most of these protocols are not adaptive to an application or system's requirements. There is a need to develop adaptive transport and routing protocols. Such protocols, for example, should reduce energy consumption by using either multi hop communication or a transmission power control mechanism, if that does not cause tasks to miss the deadline, or send data through multiple routes when the network becomes unstable. Moreover, the protocols should actively communicate with grid and mobility management systems to provide reliable and robust services.

4.9. Wireless Communication Technologies

Several wireless communication technologies, such as Wi-Fi Direct and Bluetooth, with different characteristics are available to transfer data from one node to another in an ad hoc mode. Moreover, most of the devices provide an option to use and switch between multiple connections simultaneously. To provide efficient and reliable data transmission, further work is required to investigate and adopt wireless communication technologies for mobile ad hoc grid and cloud.

4.10. Development of Simulation Environment

Existing simulators allow the simulation of either networking protocols or cloud-related algorithms and schemes. Simulators that support the simulation of both networking protocols and cloud management systems are not designed for mobile ad hoc environments. Further work is required to develop an environment in which both networking and computational cloud-related algorithms can be evaluated.

5. Selected Research Projects

5.1. Cloud Robotics

Researchers at the Social Robotics Lab have developed a cloud middleware to generate 3-D models of environments. The cloud middleware enables robots to perform simultaneous localization and mapping much faster than local onboard computers (Social Robotics Lab, 2016). SLAM refers to a technique for a robot to build a map of the environment without a priori knowledge and to simultaneously localize itself in the unknown environment.

At CNRS, researchers are creating object databases for robots to simplify the planning of manipulation tasks like opening a door. The idea is to develop a software framework where objects come with a "user manual" so robots can manipulate them (Erico, 2011).

Gostai, a French robotics firm, has built a cloud robotics infrastructure called GostaiNet, which allows a robot to perform complex tasks, such as speech recognition, advanced vision and face detection, remotely on a cloud (Gostai, 2016). The architecture of the GostaiNet system is presented in Figure 15.

5.2. Mobile Grid and Cloud Computing

Collaborative Drones: The objective of this project was to develop an aerial surveillance system that can be used in disaster management situations or military operations. The video data collected by unmanned aerial vehicles are submitted to a ground system that processes the data to detect and track objects, such as cars or persons, in real time (Lakeside Labs, 2012).

Figure 15. GostaiNet System Architecture.

SINUS: Self-organizing Intelligent Network of Autonomous Unmanned Aerial Vehicles (Lakeside Labs, 2016) is another project that aims to provide; 1) reliable aerial networking for robust multimedia streaming, 2) distributed coordination of unmanned aerial vehicle movement and task execution, and 3) system integration.

Fare-Share: Researchers at the Cyber Physical Systems Laboratory at Rutgers University are working on a project that aims to exploit the collective capabilities of nearby mobile and stationary devices to execute computationally intensive models for deriving physiological parameters and for acquiring context awareness in real time (Lee & Viswanathan, 2016). A block diagram of the Fair-Share project is given in Figure 16.

Content-Based Mobile Edge Networking: In traditional systems, soldiers must visit a battlefield camp to obtain the latest battlefield contents, because information required at the tactical edge is not immediately available. The CBMEN program aims to develop technologies to allow rapid sharing of up-to-date imagery, maps and other vital information directly among front-line units. The mobile device of a solider will generate, distribute and maintain contents at the tactical edge (Phoel, 2016).

6. Literature Review

6.1. Mobile Grid and Cloud Computing

In the literature, several architectures have been proposed to enable mobile devices to share resources either with pre-existing network infrastructure-based computing systems (Balan et al., 2003; Chun & Maniatis, 2009; Chun et al., 2011; Giurgiu et al., 2009; Mohamed et al., 2005; Rellermeyer et al., 2008, 2009; Satyanarayanan et al., 2009; Zheng et al., 1999; Zong, Nijim, & Qin, 2008) such as the cloud, or with other mobile devices using short range wireless communication technologies (Fox et al., 2008; Hummel & Jelleschitz, 2007; Piotr & Poznan, 2005; Robinson et al., 2005; Selvi, Sharfraz, & Parthasarathi, 2007; Shah, 2015; Shah & Park, 2011; Shah et al., 2012; Shilve et al., 2006).

One such architecture is proposed by Lim, Mukherjee, Lam, Wong, and See (2005), in which mobile devices are connected to a locally available powerful computer named a cloudlet through short-range wireless communication technologies such as Wi-Fi. The cloudlet is then connected to a cloud computing system through a satellite. The aim is to reduce satellite communication latency. The cloudlet system architecture is presented in Figure 17.

The object recognition application has been implemented to evaluate the performance. The solider captures and sends an image of an individual to the cloudlet, which preprocesses the image and sends it to the cloud computing system through a satellite for further processing.

Ferzli and Khalife (2011) propose a system that allows students to offload an image and video processing tasks to an internet cloud. A single hop ad hoc cloud has been suggested to compute a climate model (Busching et al., 2012). In the proposed scenario, train commuters form an ad hoc cloud comprising smart phones, which is used to forecast local weather and ozone concentrations at the destination using the data collected and transmitted by passengers on another train coming from the destination. A hybrid cluster comprising stationary and mobile nodes interconnected by wireless and wired networks is proposed by Cheng et al. (2000). The proposed system could be used by an architect equipped with wearable computing devices to execute a computationally intensive 3D rendering application. Six online and batch scheduling heuristics are proposed by Li, Pei, Wu, and Shen (2015) to offload computationally intensive independent tasks on a mobile ad hoc cloud. MinHop allocates a task to a node accessible through a minimum number of hops. The Minimum Execution Time with Communication algorithm selects the node that takes the least amount of time to execute a task, whereas the Minimum Completion Time with Communication algorithm assigns a task to the node with minimum expected completion time. Both online scheduling algorithms take into account computation and communication cost, but the former focuses on execution time, whereas the latter focuses on completion time. For batch scheduling, MinMinComm and MaxMinComm algorithms have been proposed. MinMinComm estimates the completion time of each task on each node and selects the node with the minimum earliest completion time, whereas MaxMinComm focuses on the maximum earliest completion time. The proposed scheduling heuristics are based on traditional MET, MCT, MinMin, and MaxMin heuristics and are extended to consider communication cost when estimating the task completion time. The problem of deciding whether to execute tasks locally or on a remote system is investigated by Kumar and Lu (2010). The evaluations suggest that tasks with high computation-to-communication ratio should be executed on a powerful remote system. The scheme proposed by Shi, Vasileios, Mostafa, and Ellen (2012) performs task allocations under varying assumptions regarding the connectivity environment. For example, in the ideal network environment, where the future contact can

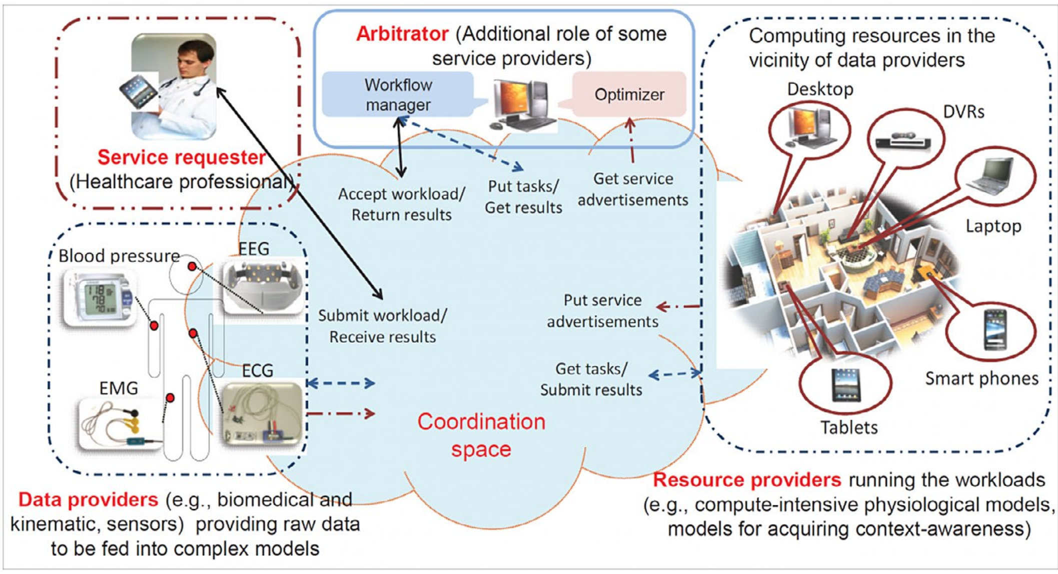

Figure 16. Block Diagram of Fair-Share Project.

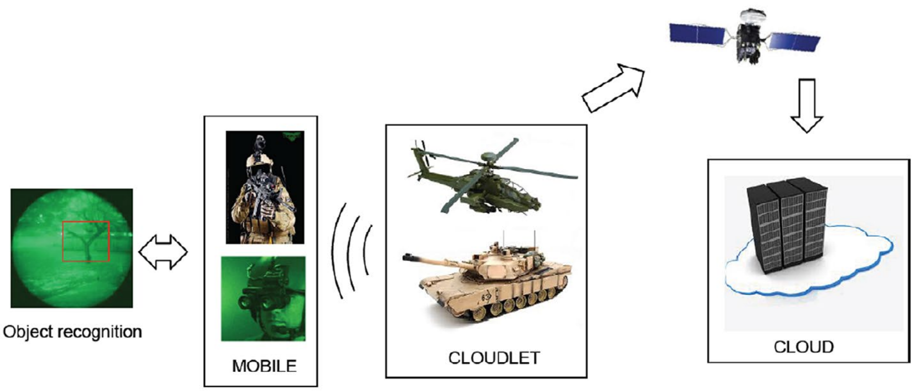

Figure 17. Cloudlet System Architecture.

be accurately predicted, the authors propose a greedy task allocation algorithm that iteratively chooses the destination node for every task with the minimum task completion time. The scheme is based on a decentralized architecture and does not provide any mechanism to address node mobility. Rodriguez, Mateos, and Zunino (2014) have proposed various job stealing techniques, such as random stealing, best rank aware stealing and worst rank aware stealing, based on a centralized architecture are developed to reduce the processing energy consumption. The rank is calculated based on a benchmark factor for each device, the estimated uptime of a device and the number of jobs assigned to that device. Simin, Mohammad, and Behrouz (2015) have proposed three offloading architectures for mobile clouds: (i) offloading to a cloud server accessible through a wide area network, (ii) offloading to a nearby server accessible through a local area network, and (iii) offloading to nearby mobile devices. The problems of unpredictable network connectivity, node mobility, energy consumption, and device failure are addressed by Hariharasudhan, Lee, Rodero, and Pompili (2014). To improve performance and energy efficiency, a context-ware mobile cloud application model is presented by Khan, Othman, Khan, et al. (2015). The proposed model uses several offloading strategies to support numerous applications. The problem of dividing tasks into sub-tasks and allocating sub-tasks to mobile nodes is addressed by Chen, Hao, Lai, and Wu (2015), whereas problem of unstable connectivity is addressed by Deng, Huang, Taheri, and Zomaya (2015). The proposed scheme introduces a mobility model and fault- tolerance mechanism, and aims to reduce execution time and energy consumption. To improve energy efficiency, a computational offloading framework is proposed by Shiraz, Gani, Shamim, et al. (2015). Compared to existing offloading strategies, proposed framework considers overhead of components migration at runtime.

To reduce transmission energy consumption, Shah and Park (2011) have proposed an energy-efficient resource allocation scheme for the allocation of tasks on a mobile ad hoc grid. The scheme is based on a hybrid architecture that results in effective allocation and reduces the processing burden from a single node and communication cost associated with the exchange of control information. To address the node mobility problem, a two-phase resource allocation scheme is proposed by Shah et al. (2012). The scheme is divided into two phases. The first phase exploits the history of user mobility patterns to select nodes that provides long-term connectivity, and the second phase considers the task and dependency types and uses the distance information among the nodes selected in the first phase to reduce communication costs. An efficient and robust resource allocation scheme for the allocation of interdependent tasks on a mobile ad hoc grid is proposed by Shah (2015). The scheme aims to reduce the task completion time and the amount of energy consumed during the transmission of data.

6.2. Sensor Cloud[1]

Several approaches have been proposed to integrate sensing nodes with grid and cloud computing systems. For example, a hybrid static and mobile grid computing system is proposed by Viswanathan, Lee, and Pompili (2012), in which mobile and static computing devices and bio-sensing nodes are integrated and presented as one unified system. The bio-sensors collect vital signs such as the blood pressure, temperature, electrocardiogram, and oxygen saturation of an individual. The collected data are processed and analyzed on a mobile grid computing infrastructure to determine the health of an individual. To address uncertainty, an idea of application waypoints has been introduced, in which the service provider executing the application task reports to the broker with an estimate of residual task completion time. If the broker does not receive feedback about the estimated residual task completion time from the service provider at the specified waypoint, it marks the service provider as failed and assigns additional resources to take over the incomplete tasks. A resource allocation algorithm to efficiently process telemedicine data in the grid is proposed by Vigneswari and Mohamed (2014). In the proposed algorithm, the sensors attached to a patient's body collect and send health-related data to the grid through a mobile device. A patient management application deployed on the grid processes and analyzes the patient's data.

The sensor-cloud infrastructure proposed by Yuriyama and Kushida (2010) integrates sensors with a cloud computing system. In the sensor-cloud infrastructure, physical sensors integrated with the cloud computing system are virtualized as virtual sensors and are provided as a service. A pull-based resource allocation algorithm is proposed by Kim, Khamra, Rodero, Jha, and Parashar (2011), in which a service provider node pulls tasks from the service broker nodes, executes them and submits results once a task completes its execution. Tham and Buyya (2005) have proposed a sensor grid platform to combine real-time data about the environment with vast computational resources. The proposed sensor grid platform can be deployed using a centralized architecture or decentralized architecture. In a centralized architecture, a sensor network connected to the grid collects data while processing of the data is carried out on the grid. In a distributed architecture, a sensor network collects data and performs simple data processing tasks while computationally intensive data processing and analysis tasks are executed on the grid.

A sensor data collection network to integrate sensor data into grid applications is discussed by Gaynor et al. (2004). The sensor data collection network includes three key components: The data collection network, sensor entry points, and application entry points. The data collection network discovers, filters, and queries multiple sensor networks. Each sensor network has one or more sensor entry points that map application data requirements onto low-level sensor network operations. The application entry points provide application connectivity to a data collection network. The proposed system is based on publish-subscribe paradigm. A sensor network publishes sensor data and metadata through service entry points, whereas an application subscribes to the sensor network and receives data in real time. Lim et al. (2005) have proposed a scalable proxy-based architecture for a sensor grid, in which nodes in a wireless sensor network are provided as a service on the grid. The use of proxy-based architecture, where the proxy acts as an interface between the grid and sensor network, supports a wide range of sensor network implementations. All of these systems and schemes, including the schemes proposed by (Chun & Maniatis, 2009; Chun et al., 2011; Cuervo et al., 2010; Garriss et al., 2008; Li & Li, 2011; Mohamed et al., 2005; Rellermeyer, Alonso, & Roscoe, 2007; Rellermeyer et al., 2008; Satyanarayanan et al., 2009; Zong et al., 2008) assume a pre-existing network infrastructure and are thus not suitable for ad hoc network environments.

7. Conclusion

Due to recent advances in mobile computing and communication technologies, it has become feasible to develop the next generation of distributed computing systems, where mobile nodes, such as robots and aerial vehicles, should be able to share resources either with pre-existing network infrastructure-based computing systems, such as the cloud, or with other mobile nodes using short range wireless communication technologies, such as Wi-Fi Direct.

Compared to traditional parallel and distributed computing systems, mobile grid and cloud computing systems pose numerous challenges due to node mobility, a shared and unreliable communication medium, low bandwidth, high latency, limited battery power, and an infrastructure-less network environment. This paper discussed applications of mobile grid and cloud computing systems, research challenges, future research directions, and recent advances in the field.

Note

1. The related work described in Section 6.2 has also been included in my paper (Shah, 2016) submitted to JATIT.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This work was supported by Hankuk University of Foreign Studies Research Fund of 2017.

Notes on contributor

Sayed Chhattan Shah is an assistant professor of Computer Science in the Department of Information Communication Engineering at Hankuk University of Foreign Studies Korea. His research interests lie in the fields of parallel and distributed computing systems and ad hoc networks. He received his Ph.D. in Computer Science from Korea University in 2012 and his M.S. in Computer Science from National University of Computer and Emerging Sciences in 2008. Prior to joining HUFS, he was a Senior Researcher at the Electronics and Telecommunications Research Institute South Korea and Engineer at the National Engineering and Scientific Commission Pakistan. He also held faculty positions at Korea University, Dongguk University, Seoul National University of Science and Technology, Hamdard University and Isra University. Shah is an Editor for the Advances in Networks and IEEE Internet Initiative, Associate Editor for the Journal of Information Processing Systems, Journal of Telecommunication Systems, and on the editorial board for the Computer Communication and Collaboration Journal. He has served as the Conference Chair and on program committees of various international conferences.

References

Balan, R.K., Satyanarayanan, M., Park, S.Y., & Okoshi, T. (2003, May). Tactics-based remote execution for mobile computing. International Conference on Mobile Systems, Applications and Services, San Francisco.

Bonaccorsi, M., Fiorini, L., Cavallo, F., Esposito, R., & Dario, P. (2015). Design of cloud robotic services for senior citizens to improve independent living and personal health management. *Ambient Assisted Living, 11*, 465–475. doi: 10.1007/978-3-319-18374-9_43.

Busching, F., Schildt, S., Wolf, L. (2012, June). DroidCluster: Towards smartphone cluster computing - the streets are paved with potential computer clusters. 32nd International Conference on Distributed Computing Systems, 114-117. doi: 10.1109/ICDCSW.2012.59

Chao, Y., Zhang, H., & Song, L. (2015). Demo: Wi-Fi Multihop: Implementing device-to-device local area networks by android smartphones. Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 405-406. doi: 10.1145/2746285.2764931

Chen, M., Hao, Y., Lai, C., & Wu, D. (2015). On the computation offloading at ad hoc cloudlet: Architecture and service modes. *IEEE Communications Magazine, 53*, 18–24. doi: 10.1109/MCOM.2015.7120041.

Cheng, L., Wanchoo, A., Marsic, I., (2000, May). Hybrid cluster computing with mobile objects. Proceeding of 4th International Conference on High Performance Computing in Asia-Pacific Region, 14-17. doi: 10.1109/HPC.2000.843570

Chun, B.G., Ihm, S., Maniatis, P., Naik, M., Patti. A., (2011, April). CloneCloud: Elastic execution between mobile device and cloud. Proceedings of the sixth conference on Computer systems, Salzburg, 301-314.

Chun, B.G., & Maniatis, P. (2009, May) Augmented smartphone applications through clone cloud execution. Proceedings of the 12th Workshop on Hot Topics in Operating Systems, Monte Verita. 8-9.

Cuervo, E., Balasubramanian, A., Cho, D.K., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010) MAUI: Making smartphone last longer with code offload. Proceedings of the 8th international conference on mobile systems, applications, and services, San Francisco, 49-62.

Deng, S., Huang, L., Taheri, J., & Zomaya, A. (2015). Computation offloading for service workflow in mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*, 3317–3329. doi: 10.1109/TPDS.2014.2381640.

Dinh, H.T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communication and Mobile Computing, 13*, 1587–1611. doi: 10.1002/wcm.1203.

Erico, G. (2011). Cloud robotics: Connected to the cloud, robots get smarter. Retrieved from http://spectrum.ieee.org/automaton/robotics/robotics-software/cloud-robotics

Ferzli, R., Khalife, I., (2011, January). Mobile cloud computing educational tool for image/video processing algorithms. Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop, 529-533. doi: 10.1109/DSP-SPE.2011.5739270

Fox, G.H., Rui, W., Isaac, K., (2008, May). A collaborative sensor grids framework, International Symposium on Collaborative Technologies and Systems. doi: 10.1109/CTS.2008.4543909

Garriss, S., Caceres, R., Berger, S., Sailer, R., van Doorn, L. and Zhang, X. (2008, June). Trustworthy and personalized computing on public kiosks. Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, Breckenridge. 199-210.

Gaynor, M., Moulton, S.L., Welsh, M., LaCombe, E., Rowan, A., & Wynne, J. (2004). Integrating wireless sensor networks with grid. *IEEE Internet Computing, 8*, 32–39. doi: 10.1109/MIC.2004.18.

Giurgiu I., Riva O., Juric D., Krivulev I., Alonso G. (2009, November). Calling the cloud: Enabling mobile phones as interfaces to cloud applications. Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware, Urbana Champaign, 1–20.

Gostai. (2016). GostaiNet. Retrieved from http://www.gostai.com/activities/consumer

Hariharasudhan, V., Lee, E., Rodero, I., & Pompili, D. (2014). Uncertainty-aware autonomic resource provisioning for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*, 2363–2372. doi: 10.1109/TPDS.2014.2345057.

Hummel, K.A., & Jelleschitz, G. (2007, May). Robust de-centralized job scheduling approach for mobile peers in ad hoc grids. 7th IEEE Int. Symp. Cluster Computing and the Grid. doi: 10.1109/CCGRID.2007.9

iCore. (2014). iCore cognitive framework. Retrieved from http://www.iot-icore.eu

Khan, A., Othman, M., Khan, N., Abid, S., & Madani, A. (2015). MobiByte: An application development model for mobile cloud computing. *Journal of Grid Computing, 13*, 605–628. doi: 10.1007/s10723-015-9335-x.

Kim, H., Khamra, Y., Rodero, I., Jha, S., & Parashar, M. (2011). Autonomic management of application workflows on hybrid computing infrastructure. *Scientific Programming, 19*, 75–89. doi: 10.3233/SPR-2011-0319.

Kumar, K., & Lu, Y. (2010). Cloud computing for mobile users: Can offloading computation save energy? *IEEE Computer, 43*, 51–56. doi: 10.1109/MC.2010.98.

Lakeside Labs. (2012). Collaborative microdrones. Retrieved from http://uav.lakeside-labs.com/overview/cdrones

Lakeside Labs. (2016). Self-organizing intelligent network of UAVs. Retrieved from http://uav.lakeside-labs.com/overview/sinus.

Lee, E. and Viswanathan, H. (2016). Uncertainty-aware resource provisioning in mobile computing grids for real-time in-situ data processing. Retrieved from http://nsfcac.rutgers.edu/cps/projects/uncertainty-aware-resource-provisioning-mobile-computing-grids-real-time-situ-data

Li, B., Pei, Y., Wu, H., & Shen, B. (2015). Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. *Journal of Supercomputing, 71*, 3009–3036. doi: 10.1007/s11227-015-1425-9.

Li, C., & Li, L. (2011). Trade-offs between energy consumption and QoS in mobile grid. *Journal of Supercomputing, 55*, 367–399. doi: 10.1007/s11227-009-0330-5.

Lim, H.B., Mukherjee P., Lam, V., Wong, W.F., & See, S. (2005, November). Sensor grid: Integration of wireless sensor networks and grid. Proceedings of 30th IEEE Conference on Local Computer Networks, Sydney, 91-98.

Mell, P.M., & Grance, T. (2011). NIST definition of cloud computing. *National Institute of Standards and Technology*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Mohamed, M., Srinivas, V., & Janakiram, D. (2005). MOSET: An anonymous remote mobile cluster computing paradigm. *Journal of Parallel Distributed Computing, 65*, 1212–1222. doi: 10.1016/j.jpdc.2005.04.012.

Mohanarajah, G., Hunziker, D., D'Andrea, R., & Waibel, M. (2015). Rapyuta: A cloud robotics platform. *IEEE Transactions on Automation Science and Engineering, 12*, 481–493. doi: 10.1109/TASE.2014.2329556.

Nick, T. (2014). A modern smartphone or a vintage supercomputer. Retrieved from http://www.phonearena.com/news/A-modern-smartphone-or-a-vintage-supercomputer-which-is-more-powerful_id57149

Piotr, G., & Poznan, B. (2005). Access from J2ME-enabled mobile devices to grid services. *Supercomputing and Networking Center*, 61–704. Retrieved from http://www.gridlab.org/WorkPackages/wp-12/res/docs/Mobility_Paper_GridLab.pdf

Phoel, W. (2016). Content-based mobile edge networking program. Retrieved from http://www.darpa.mil/program/content-based-mobile-edge-networking

Rellermeyer, J., Alonso, G., & Roscoe, T. (2007, November). R-OSGi: Distributed applications through software modularization. ACM/IFIP/USENIX 8th International Middleware Conference, Newport, 50–54.

Rellermeyer, J., Duller, M., & Alonso, G. (2009, May). Engineering the cloud from software modules. Proceedings of Workshop on Software Engineering Challenges in Cloud Computing, Vancouver, 32–37.

Rellermeyer, J., Riva, O., & Alonso, G. (2008) AlfredO: Architecture for flexible interaction with electronic devices. 9th ACM/IFIP/USENIX International Conference on Middleware, Lecture Notes in Computer Science, 5346, Leuven. 22–41.

Robinson, J., Frey, J., Clark, A., Reynolds, A.D., & Bedi, B. (2005, July). Sensor networks and grid middleware for laboratory monitoring. International Conference on e-Science and Grid Computing, doi: 10.1109/E-SCIENCE.2005.73

Rodriguez, J.M., Mateos, C., & Zunino, A. (2014). Energy-efficient job stealing for CPU-intensive processing in mobile devices. *Computing, 96*, 87–117. doi: 10.1007/s00607-012-0245-5.

Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). Case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing, 8*, 14–23. doi: 10.1109/MPRV.2009.82.

Selvi, V., Sharfraz, S., & Parthasarathi, R. (2007, May). Mobile ad hoc grid using trace based mobility model. Proceedings of the 2nd international conference on Advances in grid and pervasive computing, Paris, 274–285.

Shah, S.C. (2016). Network aware resource scheduling in sensor cloud. *Journal of Theoretical and Applied Information Technology*. Accepted for publication.

Shah, S.C. (2015). Energy efficient and robust allocation of interdependent tasks on mobile ad hoc computational grid. *Concurrency and Computation: Practice and Experience, 27*, 1226–1254. doi: 10.1002/cpe.3297.

Shah, S.C., Nizamani, Q., Sajjad, C., & Park, M.S. (2012). An effective and robust two-phase resource allocation scheme for interdependent tasks in mobile ad hoc computational grids. *Journal of Parallel and Distributed Computing, 72*, 1664–1679. doi: 10.1016/j.jpdc.2012.07.012.

Shah, S.C., & Park, M.S. (2011). An energy-efficient resource allocation scheme for mobile ad hoc computational grids. *Journal of Grid Computing, 9*, 303. 10.1007/s10723-011-9188-x.

Shankland, S. (2015). How 5G will push a supercharged network to your phone, home, car. Retrieved from http://www.cnet.com/news/how-5 g-will-push-a-supercharged-network-to-your-phone-home-and-car/

Shi, C., Vasileios, L., Mostafa, A., Ellen, W. (2012, June). Serendipity: Enabling remote computing among intermittently connected mobile devices. Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing, Hilton Head. 145-154. doi: 10.1145/2248371.2248394

Shilve, S., Siegel, H.J., Maciejewski, A.A., Sugavanam, P., Banka, T., Castain, R., …Velazco, J. (2006). Static allocation of resources to communicating subtasks in a heterogeneous ad hoc grid environment. *Journal of Parallel Distributed Computing, 66*, 600–611. doi: 10.1016/j.jpdc.2005.10.005.

Shiraz, M., Gani, A., Shamim, A., Khan, S., & Wasim, A. (2015). Energy efficient computational offloading framework for mobile cloud computing. *Journal of Grid Computing, 13*(1), 1–18. doi: 10.1007/s10723-014-9323-6.

Simin, G., Mohammad, A., & Behrouz, S. (2015). Context-aware multi-objective resource allocation in mobile cloud. *Computer and Electrical Engineering, 44*, 218–240. doi: 10.1016/j.compeleceng.2015.02.006.

Social Robotics Lab. (2016). Retrieved from http://srl.informatik.uni-freiburg.de/research

Tham, C., & Buyya, R. (2005). Sensorgrid: Integrating sensor networks and grid computing. *CSI Communications, 29*, 24–29.

Vigneswari, T., & Mohamed, M. (2014). Scheduling in sensor grid middleware for telemedicine using ABC algorithm. *International Journal of Telemedicine and Applications, 2014*, 1–7. doi: 10.1155/2014/592342.

Viswanathan, H., Lee, E., and Pompili, D. (2012, June). Mobile grid computing for data- and patient-centric ubiquitous healthcare. IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things, Seoul. doi: 10.1109/ETSIoT.2012.6311263

Wan, J., Tang, S., Yan, H., Li, D., Wang, S., & Vasilakos, A.V. (2016). Cloud robotics: Current status and open issues. *IEEE Access, 4*, 2797–2807. doi: 10.1109/ACCESS.2016.2574979.

White, C. (2013). A supercomputer in your pocket. Retrieved from http://www.charliewhite.net/2013/09/smartphones-vs-supercomputers

Wolfgang, E. (1997). Distributed system principles. Retrieved from http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf

Yuriyama, M., & Kushida, T. (2010). Sensor-cloud infrastructure: Physical-sensor management with virtualized sensors on cloud computing. 13th International Conference on Network-Based Information Systems, 1-8. doi: 10.1109/NBiS.2010.32

Zheng, H., Buyya, R., & Bhattacharya, S. (1999). Mobile cluster computing and timeliness issues. *Informatica: An International Journal of Computing and Informatics* 23. http://www.cloudbus.org/papers/mcc.html

Zong, Z.L., Nijim, M., & Qin, X. (2008). Energy-efficient scheduling for parallel applications on mobile clusters. *Cluster Computing, 11*, 91–113. doi: 10.1007/s10586-007-0044-5.