TECHNISCHE UNIVERSITÄT
CHEMNITZ

IoT

Seminar Report

Chair of Computer Engineering
Dept. of Computer Science

submitted by: Mayur Tulsibhai Lakhani
Matrikel Nr.: 485315
submission date: 22.08.2021

Supervising tutor: Prof.Dr. W. Hardt
Prof. Dr. Uranchimeg Tudevdagva
Dipl. Inf. Rene Schmidt

Abstract

The core objective of the Internet of things (IOT) is to deliver real-life physical objects with computing and communication power so that they can interact and collaborate with each other making life simpler for a good standard of living. the emergence of new technologies such as cloud computing and big data have changed the core functions of computers and the IoT. Through moving service provisioning from the cloud to the edge, edge computing becomes a promising solution in the epoch of IoT to meet the delay requirements of IoT applications, enhance the scalability and energy efficiency of lightweight IoT devices, delivering contextual information processing, and mitigate the traffic burdens of the mainstay network. However, as an emerging field of study, edge computing is still in its infancy and encounter numerous challenges in its implementation and standardization. In this article, we explore the implementation of edge computing, which exploits transparent computing to build scalable IoT platforms. Precisely, we first propose a transparent computing-based IoT architecture and pinpoint its advantages and associated challenges. Then, we present a case study to clearly show how to build scalable lightweight wearables with the proposed architecture. Some forthcoming directions are finally pointed out to promote continued research efforts.

***Keywords*—** IoT, Cloud Computing, Edge Computing, Transparent Computing

List of Figures

List of Tables

List of Abbreviations

IOT Internet of things

CC Cloud Computing

EC Edge Computing

TC Transparent Computing

FC Fog Computing

VM Virtual machine

MRBP Multi-OS Remote Boot Protocol

NSAP Network Storage Access Protocol

VMM Virtual Machine Monitor

CoAP Constrained Application Protocol

MQTT MQ Telemetry Transport

TCP/IP transmission Control Protocol/Internet Protocol

1 Introduction

The Internet of Things (IoT) leads to the treating of cleverly connected devices that will gather data from the hardware on which sensors and actuators are placed to communicate with software systems via communication protocols. This is a new approach for wireless communication [b2]. It can manage the existing internet as well as also new concepts from the future. Not only coming new technologies have more advantages which have been taken from the field of Information Technologies but also the way by which we are interacting to the world with any electronic device will be revolutionized by the IOT.i.e., Imagine the way the automatic production line, health care systems, and smart city devices working. Different types of Sensors and have different functionalities measuring the real-time data which is sent to the cloud. The intelligence system running on the cloud will analyze the data, monitoring and takes decisions without any interference of humans, and provide the best services. As the technologies are increasing rapidly the latest innovation comes with the new paradigm in implementation and transformation of business process to make much efficient, easier and to provide better solutions to the consumers. Internet of Things(IOT) devices providing the aspects to the new trends such as the digitalization of business processes, connected networks, connected processes, and new opportunities in certain fields likewise in the economy, society, corporate sectors and in our personal life [1].

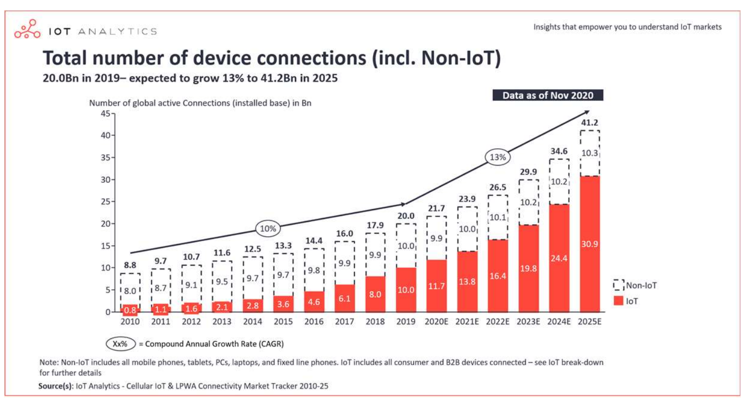

Figure 1.1: Total Number of active IOT devices [4]

Researchers are analyzing the consumption and acceptance of IoT devices from the user's point of view and expanding the use cases in daily life. However, when social media or new communication technologies arise then security and privacy issues come together with IoT[3]. As a subset of the IoT - Machine to Machine solutions are connected to devices and the internet using wireless networks [3]. According to a survey held by IoT analytics, The Number of Globally connected IoT devices are at 20 billion and it's expected to compound the annual growth rate by 12% in this number which will be 40 billion by 2025.

According to figure 1.1 there are more connections with smartphones, laptops, and computers and fixed-line. At the end of 2020, a large amount of ratio approximately 11.7 billion will be IoT devices connected to the internet and by 2025 it is expected 30 billion devices where approximately 4 devices are consumed or used per person [4].

2 Cloud Computing Paradigm

As revolutionizing the technologies, it comes up with the new business paradigm where cloud technologies have taken place for commercial computing. It provides high performance and data storage into the cloud system using web services. It is highly scalable, reliable with high performance, and can be configured for specific systems. Recent improvements in microprocessor technology and software have directed to the growing strength of hardware materials to run applications within Virtual Machines (VMs) efficiently. VMs provide both the separation of applications depending on hardware and from the other VMs, and modifications of the platform to suit the needs of the user. On demands, for the allocation of physical resources VMs rise challenges. Instead of storing in one place, the data will be stored on multiple servers [5]. Cloud computing means anything that can be hosted whether that is resources, services, or data. IoT services can be accomplished if there is universal accessibility, reliability in the context of security and privacy policies, and scalable to handle big data. Virtualization makes happen to share machine or device [6].

2.1 The confluence of IoT cloud

IoT are embedded systems that can connect to centralized or decentralized architecture. Since embedded systems have limited capabilities and to fulfill that needs the specific protocols to communicate with the software. There are many existing protocols such as MQTT, TCP socket, CoAP, AMQP, etc. There are different options to build the microcontrollers, radio modules, or peripherals which can be used by different physical networks and functionalities [7].

2.2 Communication protocols

MQTT is a message-based communication protocol that is lightweight and easier to connect with many devices. It has a broker which is intermediate software and consists of the Pub-Sub model. Subscribers receives messages from the queue implemented by the Publishers. It is designed for small cap data transfer with low network bandwidth. It is consuming less power from the IoT devices. MQTT uses TCP/IP ports with 1883 and 8883 on SSL encryption. Certificates consumes more power and as result productivity will reduce. TCP socket client connection opens on server with specific port number [8]. To create connection, server listens to the port from the client. The client needs information about “Hostname” which is the machine where the server is running. The server accepts the request from the client, and it receives the new socket connection to the same local port. If the client-side if the socket is created, then communication can be started. Communication happens by reading or writing the sockets. There are several communication connections generated between client and server. Sockets are executing on the transport layer as part of the TCP Header.

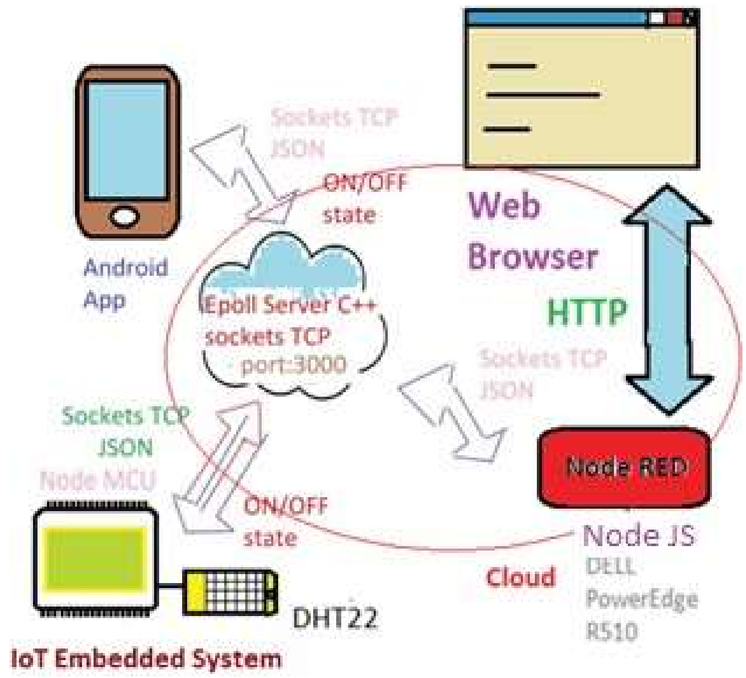

Figure 2.1: Cloud structure architecture using TCP sockets [8]

Figure 2.1 describes cloud architecture with TCP sockets. It consists of an Epoll server with C++, Node-RED, and Node JS. An Epoll server communicates with TCP sockets through IoT Embedded systems. The Node-RED is an IBM technology that accomplishes IoT server and graphical environment runs on Web browser [12]. Node JS is server technology on which Node-RED works. A mobile application connects to the Epoll server via JSON TCP sockets protocol. The HTTP protocol is used between Node-RED and Web browsers. The JSON data format message is transmitted between The Epoll server and Node-RED.

Figure 2.2 shows the architect of the MQTT protocol where the main heart is Mosquitto MQTT broker, Node-red IBM, and Node JS in Cloud. IoT devices are connected to the broker and publish the relevant topics. The customers who are connected need to subscribe to the topics to communicate with devices. The Node-RED server performs as a client. The HTTP protocol is used between a web browser and Cloud where the client manages the embedded systems and monitors consecutively temperature and other physical entities. MQTT is used for mobile app communication by connecting to the Broker as client [7].

To build the architecture a Dell Power Edge R510 server was used and the Operating system CentOS 7 was installed. Using the Wi-Fi Embedded systems are connected to the server [8]. To implement the Epoll server C++ is used and the Cmake tool to compile the code [9]. The device works like an open connection on a TCP socket and sends the message with current information to the relevant GPIO pins. The Epoll server responds to the

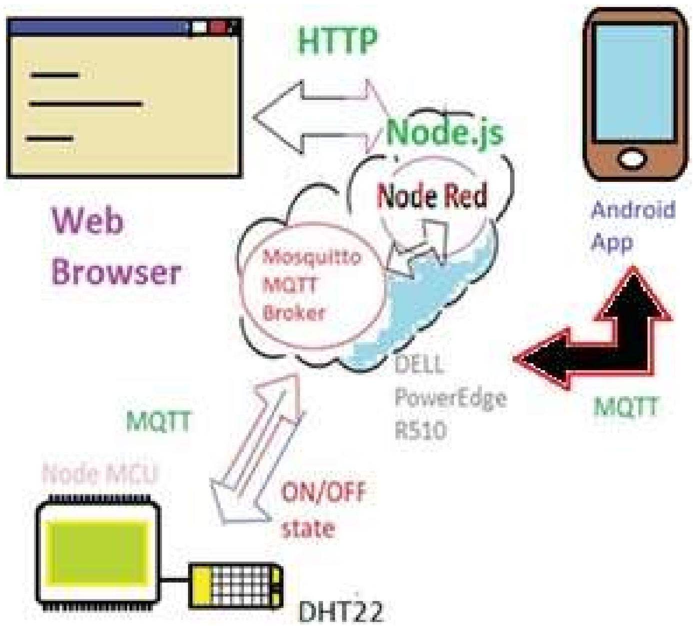

Figure 2.2: Cloud structure architecture using MQTT sockets [7]

relevant queries with minimal response time. When a mobile application connects to the cloud, GPIO sends activation command server gives a relevant message to the embedded systems and triggers GPIO pin on the IoT embedded systems [10]. The Mosquitto broker is installed on the server which acts as Pub-Sub Model. The Node-RED technology from IBM uses Node JS server technology running JavaScript for clients.

2.3 Challenges with Cloud and IoT combine Paradigm

However, there are several challenges of the Cloud and IoT combine paradigm. The data received from the IoT devices are mostly unstructured or semi-structured coming from various sensors. To provide dynamic resources management and orchestration techniques, offload from hosts to cloud dynamically. Massive data coming from different applications such as Artificial Intelligence, Image processing, or many more needs to handling in real-time processing. Major issues are as following mentioned.
1. Universal accessibility and connectivity: Access and connectivity at diverse heterogenous services/devices for large scale of users considering mobility via Application programming Interface and standards.
2. Dynamic Management/ Orchestration: users who are coming from billions of devices produce massive amounts of data.
3. Resource Utilization: Data is stored on the many servers in cloud, and which is shared and copied multiple time with different IoT apps and connected devices.
4. Personalization: It is hard to give same user experience for various smart devices or on

pc and desktops.
5. Portability: migration of the different kinds of services from the server to follow mobile users.
6. Reliability and Real-time Application: Need to communicate to device with the quick response from hard-real time applications.
7. Accountability: The different kinds of services executing on various IoT devices and consuming and storing data on different edge.

As growing data quantity exponentially data transportation became a bottle neck for cloud-based architecture [6].

3 Edge Computing Paradigm

Data is produced exponentially on the end-user terminals of the network. It would be efficient if data also can be processed on the edge network. Cloud computing has certain restrictions when the data is produced on the edge network which is already proved by the micro data center[12][13], cloudlet[14] and fog computing [15]. Edge can be defined as any computing resource or network between cloud and data i.e., a smartphone that is monitoring data from the wearable devices and send to the cloud, so it is middle between body things and clod. A smart home that handles home gadgets and the cloud.

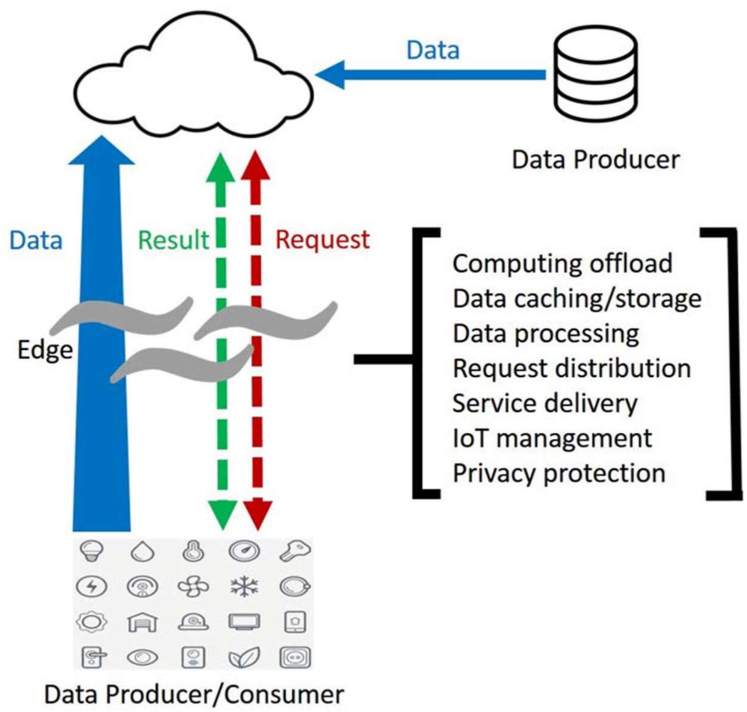

Figure 3.1: Edge Computing paradigm [17]

Figure 3.1 describes two-way directional computing where things not only consuming data but also producing the data. At the Terminals, services are not only called but also computational operations are done. So, the computations are performed offline and stored on the edge devices for cache and processing as well as distribute request and delivery service from the cloud to end-user. In this context, edge devices should be more efficient, powerful, and designed that way so it can meet all the requirements and privacy and data protection rules.

3.1 Why Do Need of the Edge Computing

3.1.1 Push From Cloud Services:

All the computational tasks are performed on the cloud have already proved one of the efficient ways of data processing, as the computational power is outstanding on the Edge. As the Technologies are emerging new paradigms such as Artificial Intelligence, Image Processing, and Big – Data Applications have produced data with a large amount of processing speed. As the result, the bandwidth of the network is important to take into account as increasing data on edge data pulling becomes a bottleneck for cloud-based technologies. Assume that Boeing 787 generates five Gigabytes of the data every second but the bandwidth of the data transmission between airplane and base station or satellite is not sufficient [18]. Another example is autonomous vehicles where one Gigabytes data is generated every second by image processing and Artificial Intelligent tools. If all data send to the cloud, then it will take a lot of time to respond. So, data need to be processed on the edge itself for a shorter response time to support more efficient processing.

3.1.2 Pulling from IOT devices:

Mostly all Embedded systems now become part of IOT which receives the data and send data to the cloud periodically. These devices have certain abilities to perform the actions and make smart decisions. So, to perform computation every time is not necessary to push the data to the cloud instead it could be done on these Terminals and that means data will be never transmitted to the cloud. It will be generated on edge of the network. The conventional Cloud Computing architecture is not sufficient for the IOT as data increases rapidly to the edge and which is too large and has led to unrelated bandwidth and computing resources usage. Privacy protection remains an issue as IOT devices perform mobility in a real-time world. IOT devices also consume a lot of energy as they have wireless communication and by performing computation offloading the process of communication will become easier and more reliable.

3.1.3 Transmission From Data Consumer to Producer:

The end devices at edge act as consumers i.e., Watching YouTube videos. Data is also produced by users by taking photos or making videos and upload them to the cloud and share via cloud services. The Image or Video Clip will be large in data bytes and consumes a lot of bandwidth to push to the cloud. Ascertain tasks need to be performed at the edge such as compression, resolution, and other safe filters before upload to the cloud [17].

3.2 The λ CoAP Architecture

The λ CoAP architecture which is a combination of the edge and Cloud Computing provides real-time interaction and reduces the latency of communication. CoAP is a Constrained Application Protocol which is a restful web transfer protocol that reduces the overhead of heavyweight protocols such as Hypertext Transfer Protocol(HTTP) and

Transmission Control Protocol(TCP) [19]. It applies RESTful Web Services on User datagram protocol(UDP). The λ CoAP architecture solves heterogeneous problems and devices can be accessed through the Internet, networking, storage, processing units with a limited amount of data by IoT devices. Over arbitrary data and arbitrary functions for real-time processing and protect a large amount of data from human errors Nathan Marz defined Lambda Architecture(LA) [20]. The main point of the LA is to provide an already computed task's view so it can reduce resultant time. The LA architecture is divided into the three layers [19]:(a) Real-time Layer: It streams the real-time data and produces the results for the precomputed view. (b) Batch Layer: It produces batch precomputed views by processing the history of data. (c) Serving Layer: It provides ideas to display and access to all generated views.

The λ CoAP architecture and LA are used for IoT and cloud computing technologies. Sensors and actuators can be created at run-time in IoT devices without reconfiguring or reboot again [21]. The LA provides analysis and actuation from the Big-Data generated by IoT devices and reduces latency. The λ CoAP design is accompanied by a Smart Gateway to connect the LA with the underlying devices and a proxy to empower HTTP intuition. Finally, a Web user interface (UI) empowers the administration and visualization of IoT information and devices. In this way distant, this system has enabled devices to work independently, but the complete processing power, which comes about in activities over the basic IoT infrastructure and information extraction is given in the cloud via the LA.

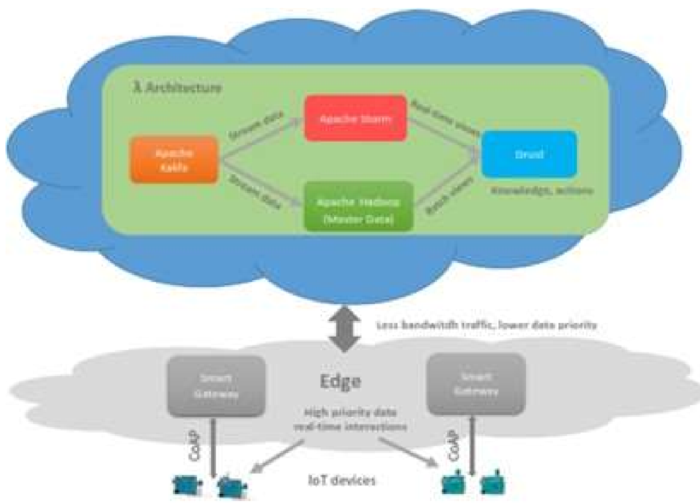

Figure 3.2: An overview of the Edge Architecture [19]

As shown in Fig (3.2) the main difference between theλ CoAP architecture and edge architecture is smart Gateway. In the λ CoAP architecture Smart gateways act as a bridge to connect with the cloud and devices. The containers are outlying with the rest and there is also the support of virtualization techniques. Docker is selected as the target IoT Gateway. Raspberry pi and rest API can be used to integrate with Web Use Interface. A web Administration portal has been used for smart Gateways to connect with gadgets and docker containers. This makes handling without cloud infrastructure and manages

Smart Gateway. The Containers have processes which is running in smart Gateways [19].

3.3 Edge-based MQTT Broker Architecture:

The First implementation is using JoramMQ v1.1, inside this the cluster broker publishes messages to the other brokers [22]. Messages are collected by all the brokers. The subscriber can receive any message if it is connected to the broker. The second implementation mentions HiveMQ which shares subscriptions with the other broker. The topic which is subscribed is the information as a subscription. The subscriptions are known by all brokers. So, each broker identifies all other brokers. When there is a scenario where clients are increasing with a large proportion their messages also will increase to the brokers as the subscription is increasing by them and this is not a good approach for broker load and network bandwidth.

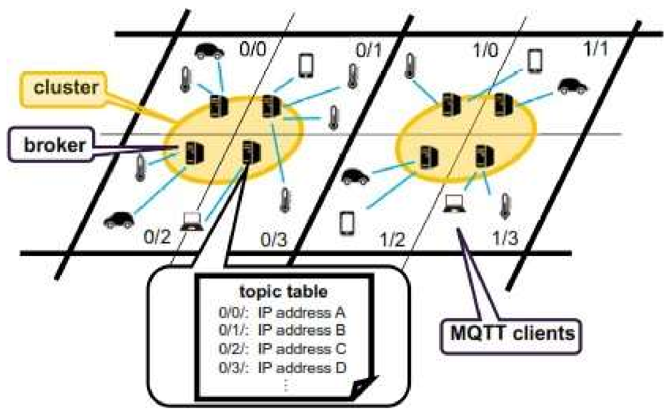

Figure 3.3: System with edge based local MQTT Broker [22]

In this system, the topic-wise group can be created for each broker. If the broker will not be available in the system, then all the topics are also not available. The Figure **??** shows the system architecture where all area is a geographical area, and it is divided into several divisions and with them also their topic which is distributed by divisions. Each area is called by Z-Ordering likewise 0 and each area contains a broker. The MQTT clients are IoT gadgets that use location-based data. Each client connects to the nearest possible broker. MQTT topics have a hierarchy that can be separated by the "/" and subscribers can use wildcards, for example for multilevel it could be "" and for single-level "+". if the topic is specifically assigned for edge brokers it will be published to subscribers. If the topic is not specified to the broker and still publishes then the broker is search for this topic and forwards it to the broker. In the end when the topic is not relevant to the particular topic then the broker will work as a proxy and send it to the subscriber. A cluster can be also generated with several brokers by nearest possible. If there is not broker available and the need to send the message then the nearest broker who has a less active client will take responsibility and publish the message to the client on behalf of the unavailable broker. From the topic table, each broker can identify another broker as well as assigned topics. So, without sharing information of clients between the several brokers it is also possible to communicate with each other [23].

3.4 Case Study

3.4.1 Cloud offloading

In the traditional cloud architecture network data was cached on the edge of the server. The data provider provides data on the Internet while IoT devices produce and consume an lOT of data. So, on edge, not only data but also operations that might need to perform is also cached. One example could be from edge computing is an online shopping application. Consider their scenario when a user comes to a search for new products and checks the detail, compares with other products, and changes settings quite often. User will put the products on cart list and changes the mind. This kind of modification regarding the shopping list or wish list is done in the cloud. Imagine on every second like thousands of users trying the same operations on this level which makes more processing time to get response and overloads to the controller. When comes to smart devices than with low bandwidth and mobile network also makes problem. The solution can be done by updating the shopping cart or wish list offline. All the cart data such as adding, modifying, or removing products can be done at the edge node and cached and synchronized with the cloud in the background [17].

3.4.2 Video Analytics

Nowadays the smartphone devices, tablets, and smart devices are used by people which is also increasing exponentially. For video analytics, these devices are used but using cloud computing is not sufficient with a large amount of data consumption and also transmission delay is long. The data is also exchanging and analyzed by the cloud servers at different distributed locations which also concern privacy issues. Consider an example where a child is lost hometown and police wants to find this child. As on each street and public center everywhere camera is placed to urban areas and on vehicles for parking or for autonomous vehicles. It could be possible that a child can be captured through this camera but the camera producing a large quantity of data that can not be published to the cloud. If data will be published to the cloud, then performing image reorganization and image detection processes will take a long time to produces this analysis. Using smartphones and devices these could be detected and searches for local camera data and only reports to the back to the cloud [17].

3.5 Edge Computing Vs. Cloud Computing [24]

There are several benefits of Edge Computing in comparison to Cloud Computing which is mentioned below in the table 3.1.

Key Features	Cloud Computing	Edge Computing
Dynamic transportation(Mobility)	Low	High
Response Time	High	Low
Storage Size	Huge	Small
Communication	Real-time	Depends on network bandwidth
Location	Remote Location	Distributed Nodes
Computation	Remote Cloud	Local edge device
Operational environment	Cloud operator control	Customer's decision
Heterogeneity	Limited	High
Computation performance	high	Medium

Table 3.1: Cloud Computing vs. Edge Computing computing [25][27]

3.6 Challenges with Edge Computing

There are several challenges that come with Edge computing as its increasing popularity of it which are mentioned below [24].

1. Reliability and fault tolerance: When the computational operations are increasing and need consistent output for hard real-time communication needs system should be reliable and without any fault give necessary output.

2. Scheduling and load balancing: For many applications such as shopping carts, smart home systems, and video analytics where the user makes a decision and needs to execute at created scheduling time. Load balancing is also the main issue while thousands of people are excessing the same resource over certain periods of time.

3.Computation and resource constraints: on Cloud computing program is written and deployed on the cloud. Cloud service provider handles which operations should be performed where users have no idea or just knowledge about abstraction. In Edge Computing, it is important to manage computation tasks and sharing of resources should be managed in a well format.

4. Remote and Resource Management: As computations are performed on the Edge server, need to check which resources are managed at the local edge server, which data need to be cached on edge of the network, and which data is most important to export to the cloud.

5.Privacy and security: As data will be published to the edge server or in the cloud it is necessary that IoT gadgets which are publishing data is quite sensitive and needs to be sent in an abstracted way that can not reveal the identity of the user, smart home, or any private details in this context.

6.Data backup: if any local server failure happens then how the data can be recoverable or if migration needs or backup on daily basis how can be possible needs to check for a large amount of the data.

4 Different computing paradigm forIoT devices [25]

From the last decades, technologies are emerging with a new computational paradigm where network and big data has created a new trend and contributed a part in industrial revolutionization. In the earlier phase, the main machine was performing all the computational tasks and storage on a single computer. In the 1980s, the demand for local area networks has grown and became popular. Now, computers can share their resources such as files and repositories to the internet which has created the client-server architecture for computing model. Mobile devices and home appliance systems have taken huge market sizes. From a user perspective, service-centric systems to develop is still an issue that came through industries and organizations. Different companies have different VMWare, Intel, and HP system processors. Virtual machine computing came to support various Operating Systems and support different application services to the single hardware unit.

The Challenges come always with new technologies here, in this case, these models require higher bandwidth of the network and need a very high-performance machine to perform computing and which in the end is not suitable for mobile devices. In this century the grid computing was proposed to share resources for multipurpose which is difficult to put for end-user. Data traffic is increasing rapidly, and Cloud Computing received a great response. It is powerful for remote computing and scalable for remote storage. For transmitting a massive amount of the data cloud server takes long round-trip latency and high bandwidth also small devices the Cloud Computing does not support hardware heterogeneity.

According to the Cisco report number of global mobile terminals will reach at 12.3 billion, and the traffic data will be consumed at 77 exabytes by upcoming year. The reason of increasing amount of data is because of Artificial Intelligence. There are several applications are used by the users such as image detection and recognition, voice recognition, and autonomous vehicles which can predict and take decision according to the situation. The network is transferring from wire to wireless communication with new technologies such as 4G(LTE), 5G etc. 5G can solve many issues regarding the network bandwidth and real time communication. It also supports low power consumption and have low latency.

Fog Computing, Edge Computing, and cloudlet have novel computing architecture which provides data transmission between network and heterogeneous systems. MEC, Fog Computing, and cloudlet perform tasks offloading schema and all the computation happens at the edge to decrease the time and improve efficiency. The motivation comes with Transparent computing which will separate the storage and computation for terminals and servers. Generally, all the software with operating systems, application programs,

Paradigms	Virtualization	Location for Computing	Location for Storage
Cloud Computing	Hyper-vision & Containers	Centralized Cloud	Centralized Cloud
Edge Computing	Hyper-vision & Containers	Edge Server	Edge Server
Transparent Computing	Meta OS	Proximal End	Transparent Server

Table 4.1: Comparison of several computing paradigms [25]

and management tools put on the server and computation will perform on the terminals. Transparent Computing points to the heterogeneity of different IoT terminals and supports cross-domain on-demand services. The Common Characteristics of Cloud Computing and Edge Computing are to provide cloud storage and decrease latency and enhance services. Transparent Computing provides complete isolation of the Computation and storage with Meta OS which uses virtualization instead of the Virtual machine and container. Here in the table 4.1 comparison of several computing paradigms is mentioned.

4.1 Virtualization

As described in the table there for different paradigms have various technologies to achieve virtualization. Transparent Computing perceives the realization using the super OS called Meta OS which creates a standard software-hardware interface to shield different hardware differences of the terminal devices. The Meta OS makes the terminals to enable to choose an OS instance on-demand and transfers the block of the code to the device. Cloud computing and Edge Computing mainly perceive virtualization using hypervisors and containers. The middleware tier is a hypervisor that manages the terminals and OSes. The various OS arrives as Virtual machines and hypervisor acts as virtual machine monitor to handle Virtual Machines and provide hardware, memory, CPU, and disk to every Virtual machine. The container techniques are small lightweight OS layer with virtualization technique that runs on any host system OS. The Containers are resource-efficient and easy to deploy. The Meta OS can be flexible to schedule and load resources from the remote servers. It provides management at the central and monitors the functions for lightweight smart terminals. Thus, it can reduce extra overhead by virtualization and increase system performance.

4.2 Location for computing

In Cloud Computing architecture computation is performed at the central cloud server. TC happens at the proximal end. Application program and instance OS loaded on the demand to the terminal for computation. Many tasks are lightweight which can be handled by the terminals. TC makes optimization of the resources so it can be easily loaded to the terminal and suits for computation. In an era of big data TC ships, the data and code to the block stream service, and without a push, all the massive collection of data makes computation on the terminals while CC has a central server for the storage of the data. Edge computing has edge servers running locally where the computation of the tasks is performed to reduce the latency of the data. CC has mainly three levels of computing service models such as IaaS, PasS, and SaaS. The nearby computing resource is handled by the edge to reduce workload it is possible to do the computation of tasks on the edge or on a cloud server.

4.3 Location for storage

TC perceives centralized resource management functions that preserve OSes, supporting tools, and application code on the backend server [25]. The front end of the TC consumes necessary protocols and a bare machine. The protocols MRBP stands for Multi-OS Remote Booting Protocol and NSAP stands for Network Storage Access Protocol are used as part of the Meta OS. Cloud Computing provides scalable storage to end-users on-demand i.e., Cloud Storage. In Edge Computing the data Stored at the edge near the data sources. It has communication overhead is low, bandwidth cost and interaction delay also low. The Edge servers transfer proximate storage resources to the users to preserve data directly at the data collection point without pushing to the cloud [25].

5 Transparent Computing Paradigm

Nowadays, smart devices are increasing towards the IoT where lightweight devices consider as smart terminals. The Network computing environment is participating with mobile internet where mobile devices are making requests to the services via wireless communication networks [27]. The cloud services and servers are accessed from the remote desktops via end users. The tasks are computed, and results are analyzed and displayed to the terminals. In server-centric computing paradigms many advantages such as centralized storage, management, easy to maintain resources and services, high utilization of server however when comes to the terminal still evolves some limitations. The terminal has limited storage capacity and needs high processing power to save as well as execute the OS and software. Eventually, it might be possible to lose control of the terminal, and data protection, user privacy, and data security will be compromised. The Terminal only shows the results from the processed computational tasks or functions which come from the cloud server. As User Experience consider then it may not satisfy the needs of the user. The architecture was first proposed in 2004 to solve these challenges [26].

The core logic behind this architecture is to store all the resources, data, software, OSes, applications, and user profile on the server. Without any hardware, the implementation user can obtain instance OS and software to the terminal. Architecture With Transparent Computing users can access on-demand service through the network with any type of device without the detail of the location of OSes, and software. No OS or application is installed on the terminal.

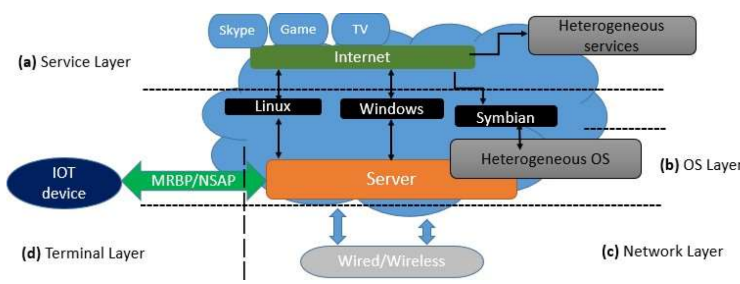

Figure 5.1: Transparent Computing Architecture [27]

As shown in Figure 5.3, the architecture is made with services, OS, network, and terminal layer. In the Services layer, the user profile and software will be saved. Different kind of heterogeneous services which will automatically be associated with the different underlying Oses. Users can request any kind of service without taking care of OSes. Services can be deployed or downloaded or installed by users with many options. The second layer is the OS layer which stores multi-OSes to support heterogeneous services, maintenance of OSes should be handled by the server. A user request to the service and particular OS is selected within the OS kernel and this kernel is transmitted with the request to the terminals. This request is completed on the terminal in a buffer-enabled block or streaming way. The third layer is the network layer which manages communications and data transfer between server and users. There are two communication protocols such as MRBP(Multi-OS Remote Booting Protocol) and NSAP(Network Storage Access Protocol) developed to make remote booting and block transmission. The Fourth layer is for receiving and executing services on different kinds of terminals such as smartphones, PCs, tablets, or any wearable device. The terminals only store underlying BIOS and selected protocols with some management tools which makes the terminal secure and easy to maintain.

5.1 MRBP and NSAP

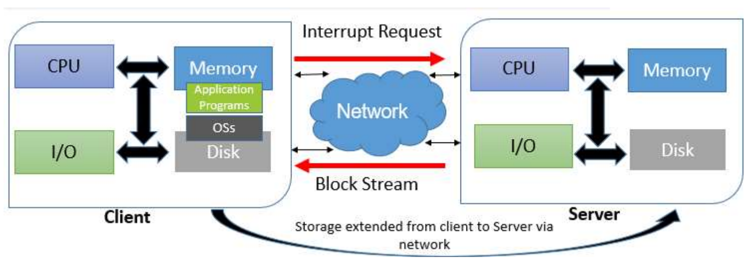

Figure 5.2: The extended von Neumann Architecture [27]

Meta OS consists of two protocols i.e., MRBP and NSAP. MRBP is mainly purposed to start the initialization of interrupt signals from the end terminals and send boot requests to the server. After searching the specific OS for the terminal, MRBP client downloads and initiate NSAP to enable virtual I/O request for the terminal. The client requires OS stored from the server via a virtual I/O device and now this OS is booted and execute in the memory of the client according to a selection of the user. The NSAP consists of assembly instruction modules that have the main purpose to transmit instructions, data, and signals to trigger I/O services between terminal and server. At last, the server includes all the services ad requires data that will be sent on demand of the user to the terminal. The server NSAP also checks for the signal whether the disk access request from the client is trustworthy while data delivery happens which can be performed using the timeout function between server and client to check the request is for validate resource access or not.

5.2 Why need to extent von Neuman Architecture?

In Cloud Computing there are two categories according to usage by the users. First Category is that where all the applications are hosted to the central data center. i.e., Google Docs, Salesforce, etc. These applications can be accessed using authentication via browser. These technologies are managing maintenance and management issues in comparison to traditional offline desktop installation. The Second category is Virtual

Machine based thin-client approach. i.e., Xen Desktop, VMware View, etc., This software makes instances of the OS and generates a virtual desktop. This virtual desktop can be excessed from any client device such as smartphones, PCs, and thin clients.

5.3 Problems of von Neumann Architecture:

As existing von Neumann Architecture, it consists of five main components such as control unit, arithmetic and logical unit, memory, and I/O. As embedded systems are small lightweight processes consume and storage capacity is also limited. It is also difficult to support graphical intensive applications. i.e., multimedia apps, video apps that consume a lot of network bandwidth. Assume their software needs to upgrade or need to install new libraries and dependencies which makes a problem. Installation of malware software and update security functions as well as management tools and routines which also makes a lot of problems to make changes. If any mechanical problem or hard disk failure happens then the data will be lost. To overcome these problems there is a need to extend von Neumann architecture in Transparent computing [28].

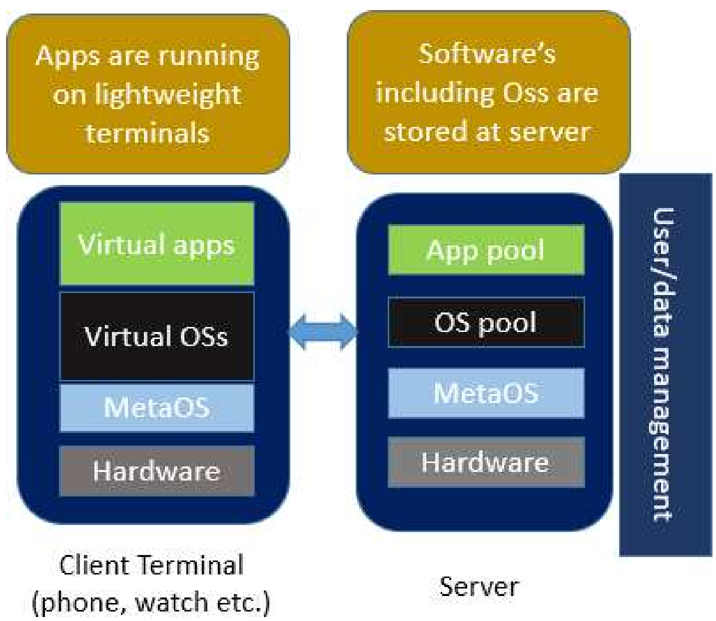

Figure 5.3: Transparent Cloud Architecture with Meta OS [27]

As shown in the figure tc-architect, The core idea of this technology is stored program procedure in networking model. In this concept, execution of the computational tasks and storage both are separated on different computers. The system or services are stored on the central server which is streamed on-demand and automatically initiated and executed to the terminals. Programs are stored on the central computer systems as opposed to the storage system. The client is responsible for the computation of several tasks and shares the programs preserved from the various computer via the network. The server consists of a repository of software and multi-OSes. The interrupt request can be generated for I/O request and storage access. The system bus has been extended from client to network. File management and user management are also possible at server [28].

5.4 Meta OS in Transparent Cloud Architecture

The existing schemes and algorithms can increase the cloud computing performance in Transparent cloud architecture. The Transparent Cloud architecture has mainly two units i.e., transparent server and the transparent client as a user. A transparent server is a cluster that consists of multiple storage instances of the OSes, application software, and user data. Meta OS runs on the server and performs mainly three tasks: (a) It creates an OS pool and manages it. (b) It creates App pool and manages it. Communication is done based on transmission protocol. Management protocols are used on the server-side to provide access control and management of the files. On the transparent terminal side, Meta OS makes remote access requests to the server, the OS booting process, and necessary applications executions done through the lightweight virtualization techniques. The transparent cloud supports heterogeneous terminals such as PCs, Smartphones, and embedded gadgets. On the Transparent user side only, Meta OS needs to install and services from the transparent server are transported to the terminals which support the OS. Advantages The Transparent Computing architecture has several advantages which are mentioned below [27]. 1. reduction in terminal's complexity and cost: Terminals can be ended up greatly straightforward, about bare machines without program, in this way making them more lightweight and diminishing equipment costs. 2. Improve User experience: Heterogeneous program from the different platform can run on the terminal because terminals are not only responsible for cross domain display results. Now these things make enable to support the various applications. i.e., video games. 3. High level of security: The path of program execution from server to the terminals can be traced as instructions can be dumped to the terminals. 4. Cross-domain capability: The application which can work across different hardware and software platform. i.e., it could be possible that Android app can run also on iOS platform. The services can be used from heterogenous software and hardware platform.

5.5 Challenges

When Transparent Computing applied to the mobile devices new challenges comes by network and new terminals. Firstly, how to support the lightweight energy consumption devices. In compared to mobile devices PCs, and laptop are more powerful in processing and computing. Therefore, the resource utilization and management should be done in careful way. In 2013 cross-domain mobile transparent computing proposed which provides on-demand resources and supporting OS and applications. As mobile devices have energy critical issues regarding battery power and due to that reason quality of services can not be compromised. Another challenge comes with the unstable and costly wireless communication network which has low network bandwidth, less transmission power. When connection established between terminals and server the data is fully transferred through wireless communications. Although current wireless technologies such as 4G and 5G technologies might be solve these issue by providing high speed data transmission but still facing challenges in user experience in mobile transparent computing. Latency is another problem in interactive response of transparent computing applications by consuming lot of resources and increasing transmission delays. To support heterogenous hardware in wireless network in mobility is still big issue. In mobile transparent computing, connectivity of hardware heterogenous wireless networks makes technical challenges where different architecture, different protocols and mobility is in various patterns [27].

5.6 Future aspects

There are many technical issues which might need to move forward for investigation to improve mobile transparent computing. Firstly, the consistent architecture needs to reduce the challenges to the terminal and network environment. This architecture should be also capable and also need to sufficient that can be collaborated with the mobile transparent computing technology. Also, it could be better that with the new technology paradigm the user experience also can be improved to provide much better services to the user. Mobile data consumption traffic is increasing rapidly which also effects to the wireless data transmission between transparent server and terminals. Other issues are regarding privacy as mobile data can be transferred to the transparent server and information regarding cookies or session can be injected to terminals which can make leakage of the privacy. If these issues can be solved in the future, then transparent computing can make new trend for the heterogeneous mobile platform and embedded terminal devices. The university of the China released smart watch called Ti-watch which can perform computation and calculation and push data to the transparent cloud which can be visible on the watch in page view. So, application runs smoothly on low-cost hardware and battery life lasts one week on normal usage [27]. To design an adaptive data transmission protocol while Transparent computing under high-speed network is still open issue. Heterogenous data collection by heterogeneous IOT devices needs to extract unstructured or semi structured data which also needed to improve Security for Transparent Computing Based IoT Platforms. From Cross platform integration for terminals and transparent servers need well-design open API interface to interact and display data on client. Transparent computing based IOT architecture is fitting for lightweight, heterogeneous, low processing IOT devices using the edge and cloud technologies. Edge servers can be used to provide enhanced security for transparent terminals.

6 Summery

To get information of administrations, lightweight IoT gadgets are getting to be the overwhelming terminals. The irregularity between the limited capabilities of lightweight IoT gadgets and increasing user requests, in any case, remains a major object to IoT application development. In this paper We have started from introducing IoT technologies and usage of it in current trends. The Cloud Computing architecture discussed which provides hosting of services and scalable data storage. However real-time communication is not possible so to overcome that problem Edge Computing paradigm discussed which provides computation on nearest edge server and data can be stored or necessary data can be uploaded to the cloud also real-time communication is possible with IoT devices. We compare Edge and Cloud computing architecture. As hardware heterogenous support comes to solve that issue we introduced the transparent computing.

Bibliography

[1] Tűrkeș, M.C., Căpușneanu, S., Topor, D.I., Staraș, A.I., Hint, M.Ș. and Stoenica, L.F., 2020. Motivations for the Use of IoT Solutions by Company Managers in the Digital Age: A Romanian Case. Applied Sciences, 10(19), p.6905.
[2] gsma.com, ”Understanding the Internet of Things (IoT), July 2014” July 31,2021. [Online] Available: https://www.gsma.com/iot/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf
[b3] www.internet-of-things-research.eu, “Internet of Things -converging Technologies for smart Environments and integrated Ecosystems”. July 31,2021. [Online] Available: http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf
[4] iot-analytics.com, “state of the IOT 2018: Number of IOT devices now at 7B – Market accelerating”. July 31, 2021 [Online] Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/ [Online]
[5] Wu, J., Ping, L., Ge, X., Wang, Y. and Fu, J., 2010, June. Cloud storage as the infrastructure of cloud computing. In 2010 International Conference on Intelligent Computing and Cognitive Informatics (pp. 380-383). IEEE.
[6] Biswas, A.R. and Giaffreda, R., 2014, March. IoT and cloud convergence: Opportunities and challenges. In 2014 IEEE World Forum on Internet of Things (WF-IoT) (pp. 375-376). IEEE.
[7] Nikolov, N., 2018, September. Research of the Communication Protocols between the IoT Embedded System and the Cloud Structure. In 2018 IEEE XXVII International Scientific Conference Electronics-ET (pp. 1-4). IEEE.
[8] STMicroelectronics, “Wireless connectivity for IoT applications”, 2015
[9] Robert Love, “Linux System Programming”, 2007 O'Reilly Media
[10] developer.android.com, “Developer Guides Android”, 2018
[11] R. Preethika , G. Prabhu, A. Prabhu, “AN ANALYSIS OF COMMUNICATION WITH IOT DEVICES USING WEBSOCKETS”, 2017
[12] M. Lekic, G. Garadasevic, “IoT sensor integration to NodeRED platform”, East Sarajevo, Infoteh-Jahorina, 2018
[13] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, 2008.
[14] E. Cuervo et al., “MAUI: Making smartphones last longer with code offload,” in Proc. 8th Int. Conf. Mobile Syst. Appl. Services, San Francisco, CA, USA, 2010, pp. 49–62.
[15] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” IEEE Pervasive Comput., vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
[16] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of things,” in Proc. 1st Edition MCC Workshop Mobile Cloud Comput., Helsinki, Finland, 2012, pp. 13–16.
[17] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L., 2016. Edge computing: Vision and challenges. IEEE internet of things journal, 3(5), pp.637-646.
[18] Boeing 787s to Create Half a Terabyte of Data Per Flight, Says Virgin Atlantic. Accessed on Dec. 7, 2016. [Online]. Available: https://datafloq.com/read/self-driving-carscreate-2-

Bibliography

petabytes-data-annually/172
[19] Fernández, C.M., Rodríguez, M.D. and Muñoz, B.R., 2018, May. An edge computing architecture in the Internet of Things. In 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC) (pp. 99-102). IEEE.
[20] N. Marz and J. Warren, Big Data: Principles and best practices of scalable realtime data systems. Manning Publications Co., 2015.
[21] C. Mart´in, M. D´iaz, and B. Rubio, "Run-time deployment and management of coap resources for the internet of things," International Journal of Distributed Sensor Networks, vol. 13, no. 3, 2017
[22] JoramMQ, http://www.scalagent.com/IMG/pdf/JoramMQ MQTT white paper.pdf
[23] Kawaguchi, R. and Bandai, M., 2020, January. Edge based MQTT broker architecture for geographical IoT applications. In 2020 International Conference on Information Networking (ICOIN) (pp. 232-235). IEEE.
[24] Naveen, Soumyalatha, and Manjunath R. Kounte. "Key technologies and challenges in IoT edge computing." 2019 Third international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC). IEEE, 2019.
[25] Zhang, Yaoxue, et al. "Transparent Computing: Development and Current Status." Chinese Journal of Electronics 29.5 (2020): 793-811.
[26] Y.X. Zhang, "Transparence computing: Concept, architecture and example", Acta Electronica Sinica, Vol.32, No.12A,
pp.169-174, 2004. (in Chinese)
[27] Y. Zhang, K. Guo, J. Ren, Y. Zhou, J. Wang, and J. Chen, "Transparent Computing: A Promising Network Computing Paradigm," in Computing in Science & Engineering, vol. 19, no. 1, pp. 7-20, Jan.-Feb. 2017, doi: 10.1109/MCSE.2017.17.
[28] Zhang, Yaoxue, and Yuezhi Zhou. "Transparent computing: Spatio-temporal extension on von Neumann architecture for cloud services." Tsinghua Science and Technology 18.1 (2013): 10-21.