Collaborative Computing Paradigms: A Software Systems Architecture for Dynamic IoT Environments

Prashant G. Joshi and Bharat M. Deshpande
Department of Computer Science & Information Systems, BITS Pilani K K Birla Goa Campus, Goa, India

Keywords: Collaborative Computing Paradigms, IoT, Computing Paradigms, Cloud Computing, Edge Computing, Mist Computing, Software Architecture, Architecture Models, System Software Architecture, Collaborative Computing, Collaborating Paradigms.

Abstract: Connected systems are omnipresent, are used to monitor and control remotely, collect data and information. A variety of software systems architectures are designed which exploit computing paradigms – Edge, Fog, Mobile and Cloud – that process and analyse the data. Such an analysis is pivotal in decision making to increase operational efficiency. IoT has transformed industries like logistics, healthcare, industrial automation and agriculture and continues to refine decision making process ultimately to enhance the systems operations and efficiency. Expanding service capability of systems, has been topic of academic research and, rests on the foundation of bringing all resources in unified resource pool and make different computing facilities collaborate. Making the different computing facilities to collaborate to realise this has been part of many theoretical and experimental studies. Industry applications have adopted such systems architectures that has enhanced the applications capabilities. This paper proposes a collaborative and unified method of system software architectures, for IoT environments, that leverage collaboration among computing paradigms. With a view to expand services, as and when needed, a unified, dynamic and distributed analytics software systems architecture was explored and experimented. Proposed collaborative method is validated through its application for vehicle and driver behaviour and data center cooling systems.

1 INTRODUCTION

Ubiquitous deployment of smart and connected devices has led to an exponential increase in the need for processing the collected data. The trend of applications based on collected data has been on a significant rise and is expected to continue growing at a substantial rate. The collected heterogeneous and high-volume data, gathered at frequent intervals, is subsequently subjected to extensive processing, enabling the derivation of variety of analytics crucial for decision-making (Donno and et al., 2019). This pervasive deployment has resulted in various successful implementations, particularly evident in areas such as vehicle tracking, where in-depth analysis of vehicle and driver behavior has been accomplished (Malekian and et al., 2015) (Türker and et al, 2016) . All such systems are built around a generic software systems architectures where the smart devices sense and enable data collection which is then processed by transmitting upstream to Cloud servers, for computing, over various networking interfaces.

In most real-world applications, it is necessary for collected data to be processed faster and results of the processing, which may be alerts or notifications, transmitted quickly. Such applications find the typical architectures in which device transmitting data to cloud for processing and generating alerts to be cumbersome. Edge computing paradigm is a paradigm of choice for many such real-world applications. With introduction of edge computing, typically in an hierarchical or layered approach along with the cloud, the computing needs for use cases are typically fixed at design time, which separates the processing into what is computed at the edge and what is done on the cloud. As discussed in (Kaushik and Naik, 2023), the temperature is sensed and transmitted to the cloud server where the data is processed and the necessary parameters are communicated back to the edge to control the temperature by switching the air conditioners.

For most applications, Cloud computing (Peter and Timothy, 2011), which provides a cost-effective, scalable and always available infrastructure for computing, dependable and managed software, elastic

Joshi, P. and Deshpande, B.
Collaborative Computing Paradigms: A Software Systems Architecture for Dynamic IoT Environments.
DOI: 10.5220/0012473000003645
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2024)*, pages 297-306
ISBN: 978-989-758-682-8; ISSN: 2184-4348
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

storage, and resource virtualization, is used. Over time the demand for computing has grown significantly with the increase in the volume of heterogeneous data collected at high frequencies from variety of devices, and real-time responses, thus giving rise to newer paradigms - Edge, Mist, Fog and Mobile - which have been used to overcame some of the limitations of Cloud computing (Iorga and et al., 2018). Every paradigm that emerged focuses on data processing closer to the source or data or the user, thus addressing the challenge of near real-time response times, interactive and on-line applications.

Software systems architectures of today, as discussed in (Garcés and et al, 2021), involve numerous interconnected components such as hardware, software, networks and users of the system. Past years have seen a very significant advancement in communication and computing technologies, that have led to a transformative impact on computing, storage, wired and wireless networks, and application deployment. Such an impact is observed across domains like consumer, commercial and industrial.

With the increasing complexity of applications, processing in real-time and faster decision making, the standardization of architectures has become a necessity. Both academia and industry have developed their own reference architectures to address this need. Academia has primarily focused on characteristics of reuse and generating domain specific knowledge (Garcés and et al, 2021). On the other hand, the industry has created reference architectures to deliver systems and solutions tailored to specific applications with a certain level of maturity. Over time, the industry has developed reference architectures for various applications and gathered best practices and references for designing new systems in different domains.

Irrespective of whether the architectures originated in academia or industry, all software system architectures have leveraged and exploited the available computing paradigms, and industry driven applications and real-world applications have been developed using them.

Paper is organised as follows. Section II provides a brief on the Computing Paradigms Landscapes which compares the current paradigms. Section III identifies the Challenges and Issues in the IoT environments summarising them based on the published research, Section IV proposes a Collaborative Computing Architecture to overcome the challenges and issues with a focus on dynamic IoT environments. Section V describes and discusses the applications and use-cases highlighting the characteristics of the proposed Collaborative Computing Paradigm. Section VI provides a conclusion and Section VII closes with the future work.

2 COMPUTING PARADIGMS LANDSCAPE

Cloud, Edge, Fog, Mist and Mobile computing have been proposed, which are characterised by variety of interfaces and access technologies, and varying capacity of resources like processing power, storage, interface and closeness to the source.

In the TABLE 1, a summary of comparison of these characteristics for various computing paradigms, considered for this research, has been shown.

Additionally, the TABLE 2 summarises the strengths and weaknesses of each computing paradigm, which is useful and typically considered when choosing one or more computing paradigms for a particular application or a use-case.

2.1 Computing Paradigms Collaboration - Challenges & Issues

Computing paradigms differ significantly based on infrastructure, storage, computing capability, access technology, resource management and service provider. It is clear from the literature as well as from the origin of the computing paradigms that different paradigms are designed for different services. For all of the computing paradigms, theoretical studies often ignore their differences and focus on their strengths put to use in realising an application.

Collaboration among the paradigms will require resources of one paradigm to be shared with other paradigm, which is not easy and it will then be difficult to converge these resources in to a unified pool. Studies show mechanism of virtualisation of each of the paradigms to such a level that one paradigm can collaborate with the other and can be an effective way of collaboration among paradigms. Such a technique is primarily designed to shield the heterogeneity among the computing paradigms. (Cai and et al, 2023; Nascimento and et al, 2023; Lewandowski and et al, 2020)

In this paper, we propose to leverage the functionality provided by each of the paradigms, use the heterogeneity with a view to enable the high priority use-cases of the application.

Table 1: Comparison of Computing Paradigms.

Computing Paradigms	Salient Features
Cloud (CC)	Provides nearly unlimited and on-demand capacity of computing and storage, thus services are scalable and expandable. All resources are available over standard network interfaces (Peter and Timothy, 2011).
Fog (FC)	Provides computing, storage and network connectivity services in close proximity to data sources and users. Thus, has a lower latency and enhances reliability in comparison to cloud computing (Iorga and et al., 2018).
Mist (MiC)	Provides lightweight processing and limited storage closer to the sensors and smart devices. Mist computing devices are based on microprocessors or microcontrollers and are developed to augment processing capabilities reducing the load on Fog and Cloud computing (Escobar and et al, 2022).
Edge (EC)	Provides functionalities of the Cloud to the edge in form of mini clouds which have the capability of transfer of data, it's processing and storage at the edge of the network. Analysis is done in real-time without latency and facilitates quicker data processing and content delivery (Fernández and et al, 2018).
Mobile (MC)	Portable devices with limited computing capability such as Mobile phones, tablets, and wearables. MC nodes can move across the environments and can feature multiple connectivity interfaces (Mahmoudi and et al., 2018).

Table 2: Computing Paradigms - Strengths and Weaknesses.

Paradigm	Real-Time	Processing Power	Storage Capacity	Scalability	Interfaces
Cloud	Low	High	High	High	Low
Fog	High	Moderate	Moderate	Moderate	Moderate
Mist	High	Moderate	Moderate	Moderate	High
Edge	High	High	Low	High	Moderate
Mobile	Moderate	Low	Low	Low	Highest

3 INTERNET OF THINGS (IoT) ENVIRONMENT - CHALLENGES AND ISSUES

IoT environments are highly demanding and require a dynamic model of software system architecture. In applications like autonomous vehicles, smart cities, agriculture, healthcare, large amounts of heterogeneous data from variety of sensors is collected. This data needs to be effectively processed to fulfil the demands of the application and stored for future use. All IoT environments require end to end connectivity with the devices and all participating components to provision, upgrade and manage the components.

Applications today still are dominated by an IoT device (IoTD) and cloud computing paradigm. Devices use network interfaces to transmit data to the cloud, where the data is stored and analysed. With the need for overcoming the challenges of real-time, heterogeneity and low latency proposed in (Bonomi and et al, 2012) Fog Computing, a high virtualised platform which is spatially close to the devices and provides compute, storage and networking services between the devices at the edge and the cloud computing core. The paper discusses fog computing and the IoT in areas of connected vehicle, smart grid, and wireless sensor and actuator networks. This paved way to build a substantial body of work with fog computing at its' core. An interplay between fog and cloud is described, but not developed any further in the paper.

Fog computing has been classified and challenges summarised in (Carla and et al., 2019) and architectural challenges to make the system ultra-responsive have been identified. Authors have indeed done a wide and deep survey of the research work and have classified the same in major areas Architecture and Algorithm. A set of evaluation criteria like heterogeneity, QoS management, scalability, mobility, federation and interoperability has been summarised and all the classified architectures and algorithms have been evaluated based on these criteria. Architecture based classification has been further broken into end-user application agnostic and application specific while the algorithms have been further classified into computing, content storage and distribution, impact in energy consumption, and specific end-user algorithms. Authors observe that there needs to be focused and deep research in the area of federation of Fog/Edge and Cloud. Furthermore, a complete interplay of IoT, Fog, cloud and Mobile computing has not yet been attempted.

In the paper (Vasconcelos. and et al., 2019) an algorithm is proposed to make a choice between cloud, fog or mist computing based on cost, bandwidth and latency criteria. Simulation results suggest that the algorithm choice is adequate. However, this is still a choice to be made at design time. This area has been researched with proposals for workload engineering, using OpenStack-based middleware, described in (Giovanni and et al., 2019) along with the advantages of workload engineering. As the proposed solution is using software at various levels, the processing exhibits software-defined characteristics which definitely show the promise of such techniques. Viability of the solution is evaluated based on the case study on an intelligent surveillance system, but does not detail out the other aspects of orchestration policies, security, and strategies which are left for future work. A unified model for Mobile-Edge-Cloud continuum which uses Function-as-a-Service in order to bring the computation in form of micro services and an adaptive resource allocation for computing offloading has been discussed in (Baresi and et al., 2019), where edge servers are modelled as a set of linear systems.

A collaboration between cloud, fog and IoT and its effects on performance has been detailed in (Mohammad and et al, 2018). Authors identify the integration issues within Cloud-IoT. Further they evaluate the use of fog and mobile computing in collaboration with Cloud-IoT. Fog and Cloud have been compared using metrics like processing delay, processing cost, processing capability and task length. An entire test bed was used to simulate the scenarios and analyse the results. Observations have been that Fog does reduce processing delay to a significant level however there are limits to the efficiency. With their research they have identified the future areas of finding the synergy in the collaboration, scalability - horizontal and vertical, and collaboration with mobile computing.

As large amounts of data is gathered even in sub-second intervals (connected vehicle and wind power) the existing architecture and infrastructure is not enough to take the load of transmitting the data to core cloud and analysing the same. One of the key research for integration for all of the IoT, cloud/edge is presented in the (Munoz and et al, 2018) which proposes software defined networking to arrive at the best possible distribution of load across cloud or edge. It builds an orchestration architecture among the components. The authors suggest the distribution of analytics between the core Cloud and the Edge of the network to efficiently utilize network resources and enable the deployment of dynamic and efficient IoT services. However, while this technique demonstrates success, it does not leverage the potential benefits of Fog and Mobile computing architectures alongside Cloud and IoT. A collaborative and unified architecture will be able to achieve more dynamic and efficient analytics by distribution of analytics and efficient use of the network resources.

The TABLE 3 summarises the challenges/issues in IoT, Cloud, Fog and Mobile computing integration.

Every IoT software system is characterised by elements of compute, communication, sense/collect data, store data and user input/output, which is described in TABLE 4.

4 COLLABORATIVE COMPUTING PARADIGMS - METHOD OF SYSTEM SOFTWARE ARCHITECTURE

To provide a method to overcome the challenges identified in Computing Paradigms and IoT, a three-prong approach is proposed.

- First, IoT solutions need to be built on an architecture of infrastructure which can enable and create opportunities for enabling the dynamism and expansion of services
- Second, access of the components of the system – both infrastructure and software systems – will need to be built on well-established standards that enable ease of operation and interoperability
- Third, on such an infrastructure software system are to be built which can bring the functional dynamism

Considering that the typical system architectures involve sensing one or more parameters and transmitting to the edge or cloud processing; the layered or hierarchical approach is the first one which will need to change. With a notion that each of the computing paradigms, in their heterogeneity, when made available for a use case in a collaborative manner, have the capability to enhance the system completely. Here the proposed solution is to have a completely interconnected system which has all computing paradigms accessible to the other – a total interconnection which can enable a dynamic interplay.

Table 3: Challenges & Issues in IoT, Cloud, Fog, & Mobile Computing Integration.

No.	Challenge/Issue	Description
1	Data Volume, frequency and Processing	Exponential increase in devices on the field lead to large volume of data collected at high frequencies. Data so collected requires adequate processing, storage and analysis capabilities (Carla and et al., 2019).
2	Software System Architecture	Developing a software system architecture that enhances the the computing capabilities needed for the dynamic IoT environments (Carla and et al., 2019).
3	Data Fusion and Dynamic Computing	In dynamic IoT environments, heterogeneous data is collected using numerous sensors. Fusion of such data to form relevant pieces of information for further processing.
4	Seamless and Multi-interface Connectivity	End-to-end connectivity with every IoT device is required for data collection, device upgrades, and management. Such connectivity has to be maintained through out the life of the system (Carla and et al., 2019).
5	Fog/Edge Computing and Interplay with Cloud	Computing paradigms like Edge/Fog are designed to reduce the processing delay and enable faster response times as close to the source of data and users. Various algorithms have been explored for interplay and federation of Fog/Edge and Cloud, however a deeper exploration of the techniques is required (Bonomi and et al, 2012).
6	Choice of Computing Paradigms	In practice the decision for choice of computing is largely done at design time. Orchestration policies and algorithms have been developed, however they lack the dynamism required for IoT environments (Giovanni and et al., 2019).
7	Unified Model and Resource Allocation	Function as a service mechanism has been explored to enable a Mobile-Edge-Cloud continuum. However, orchestration policies and strategies for efficient resource utilisation are not explored fully. (Baresi and et al., 2019).
8	Integration and Collaboration	While integration of Cloud, Fog, and IoT has been considered along with Mobile computing, it needs further exploration, especially to bring synergies among the participating components and scalability (Mohammad and et al, 2018).

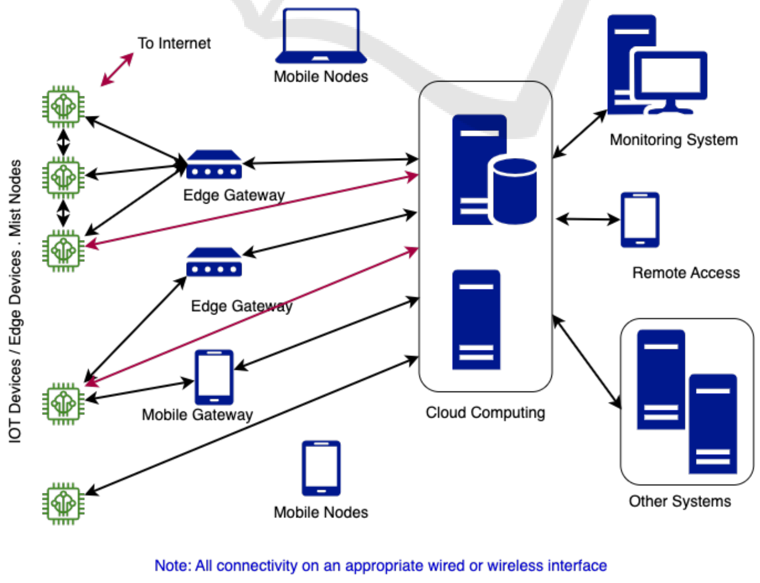

Figure 1: Collaborative computing paradigms - systems architecture.

Table 4: IoT Software Systems Architecture Characteristics.

IoT Architecture Characteristics	Description
Compute	Processing the data collected for variety of use cases. Processing is accomplished on one or more computing paradigms
Communicate	Data and commands across the system over multiple connection interfaces and protocols.
Sense/Collect Data	Sensors and devices sense and collect data and use the available communication interfaces and protocols to transmit the data.
Store Data	Collected data (raw), computed parameters and processed data is stored in the system for use at various time intervals
User IO	With a user being an element of the architecture, the system ultimately is required to serve the user and is required to provide a mechanism to interact.

Each of the paradigms then will be able to collaborate with the other and in event of issue with non-availability can use alternative approaches to get the functionality completed. An MiC can collaborate with CC without having to go through the other layers. Example: In case of an alert for a fire safety system the edge computing paradigm can itself send in the required notification (via email, an SMS or a push message sent to a web portal or a mobile device).

With the interplay enabled by interconnection, the processing of data can be dynamically distributed across other paradigms. Such a distribution can be done using well established mechanisms and algorithms or using machine learning. This can enable the fluidity of computing where allocation of tasks is dynamic and assignment is based on suitability and availability of the computing paradigm in that context. Let's take a case when a model created using machine learning and large data can only be executed using cloud server thus, any model updates shall require the cloud; however continuing functional use case can be accomplished by other available paradigms. An edge device can utilise the computing capability of Edge computing as well as distribute the part to cloud to ensure that the system is always functional. Example: Model to control the air-conditioning may be computed by the cloud; thus the data required for that shall be transmitted to the cloud; while the control of the switching of the air-conditioners can be done at the edge using mist computing.

Data transmission among the participating paradigms will become necessary for ensuring that the participating paradigm has all the necessary data and information to act on that data. Thus, input data and processed information shall need to be stored. Overall requirement of most systems shall be to have a data store for maintaining the history of collected input data and processed information which can be achieved by using cloud storage, in the long term, while during the functioning of the system the data may be distributed and at frequent intervals the same is then collected at a central location. Example: Alerts of speeding for the driver may be generated by using edge computing on the vehicle gateway, and the data can be sent – opportunistically – to the cloud later. Such data at the cloud can further be used for processing of all alerts over a period of time to ascertain a pattern of speeding alerts for that driver.

As and when required, each of the computing paradigms can be augmented by addition of more infrastructure thus providing scalability and extendability for each paradigm. This can be accomplished dynamically or as per the requirement. Example: An increase in the amount of temperature sensors in the data centers, the edge computing may need to be scale the processing capability at the edge.

With the foundation of a systems architecture of collaborative nature, the software can exploit the available computing paradigms based on the extent of dynamism it can support like in distributing the tasks across paradigms.

Characteristics of the Collaborative computing paradigms software system architecture are detailed in TABLE 5. Each of the challenges and issues listed in the TABLE 3 are mapped to the proposed Collaborative characteristics.

5 COLLABORATIVE COMPUTING PARADIGMS - APPLICATIONS AND USE-CASES

In this section two applications are discussed - Data Center Cooling System and Vehicle and Driver Behaviour System. Each application presented here is depicted using the Software Systems Architecture without and with the proposed collaborative computing approach. A comparison, based on characteristics

Table 5: Collaborative Computing Paradigms- Software Systems Architecture Characteristics.

Collaborative Characteristics	Description	Maps to Table
Inter-connection and interplay	Overcome the limitations imposed by layered approach by interconnection and interplay between smart devices and various computing paradigms.	5,6,7,8
Dynamic distribution of data processing	When not bounded by strict layering and computing distribution either at design time or by a fixed algorithm, the dynamic distribution provides increased flexibility and efficient distribution. With such approach the overall available computing capacity is increased.	1, 3
Fluidity of computing across paradigms	Allocation of tasks is dynamic and assignment is based on the suitability of a paradigm, in the context, based on the requirements and available resources at that time.	6,7,8
Storage and data management across participating paradigms	All participating paradigms will require the necessary data for processing and thus, the data will be distributed across paradigms. This data, both input data and processed data, will need to be collected and transmitted to a single store of data, typically on Cloud storage.	1
Scalability and extendability of the architecture	The proposed architecture is designed to be scalable and extendable. It supports the integration of newer devices seamlessly, allowing the architecture to grow and accommodate additional components to monitor subsystems, analyze data for predictive maintenance, and ensure the scalability and reliability of the overall system.	2,4

of Collaborative computing, identified in TABLE 5, is provided for each application.

5.1 Data Center Cooling Systems

Data centers have a requirement to maintain the temperature for optimal functioning of the servers and other network equipment. A typical installation has servers and equipment installed in racks and data centers have multiple racks. Maintaining of the temperature in the acceptable range (typically 22 to 25 deg Celsius) is accomplished using air conditioners (ACs). At all times the goal is to provide optimal cooling with lower consumption of electricity. With this goal in mind, temperature and electrical consumption are monitored on regular basis.

5.1.1 Case (1) AC with IoT (Classical Layered Approach) (Kaushik and Naik, 2023)

IoT devices for sensing temperature and humidity are installed on server racks and in the data center at the desired locations. The energy consumption for each of the ACs is monitored using smart energy meters and the remote control for air conditioners is made possible by smart devices. In this case we consider the ductless ACs. Temperature, humidity and energy consumption is collected at regular intervals (typically 10 to 30 min) along with time stamps for each of the readings. This data is then transmitted over a network interface, wired and/or wireless, to the Cloud server which pre-processes the data and stores in a database. Typically same or another server performs the computation, by accessing the data over the network from the cloud data store. This computation and processing is performed to build a cooling model (a Machine Learning (ML) algorithm). Based on such a cooling model, commands for switching the ACs are transmitted over the network to the smart device. Outcome of such a system is results in the switching of ACs, the operating hours life of the system is extended and electricity consumption is reduced. Number of sensors may be added to improve the cooling model which will increase the Cloud server load with additional data collection, storage and processing.

In regular operations the overall ACs can be monitored for healthy operations and any faults may be communicated to the cloud server and processed to create a series of alerts and notifications which can be sent over channels like email, SMS or notifications, using web-sockets, on the portal.

With the switching of ACs controlled by the cloud servers, in case of network connectivity issues, the commands for switching the ACs will not reach the smart device and the operations are at risk. While fail safe may be built for ACs to continue operation in such a scenario; overall operations will lead to inefficient consumption of electricity by the ACs as the switching will be affected. In such a scenario alerts or notifications also will not be generated and thus, the only way to monitor the system shall be manual by physical presence at the the data center.

5.1.2 Case (2) AC with IoT (Collaborative Computing)

With the IoT devices for sensing and control in place, same as in Case (1), the smart device, used for switching is made capable of storing the schedule of switching of the ACs in an autonomous manner. In this case the computation of the cooling model can be done on the cloud server based on the data collected and the switching duty cycle can be communicated to the smart device which shall operate the ACs in an autonomous manner. Thus, reducing the need for the cloud server to be actually responsible for the control of switching of the ACs. Further, an edge gateway, which is capable of computation and control for the entire data center, can be installed, which can reduce the computation and control burden from the cloud servers. This will lead to a system where the edge devices take over the prime responsibility, Cloud server is freed up to compute the model. Cloud server's resources, which otherwise were responsible for the switching, can now be used to build a better model and even compare with the data from other data centers. Edge device and Cloud server can establish an interplay and collaboration per the increase in the data center load or amount of data collected as more sensors get added for betterment of the cooling system. A mobile phone, with an app, can be used to remotely monitor the system and receive alerts. Figure 2 depicts the proposed Collaborative software system architecture for Data Center Cooling System.

In the event of network connectivity issues with the cloud server, the entire operation can still continually smoothly with the autonomy of the smart devices and the edge gateway. In event of fault, the alerts and notification can be generated and transmitted by the edge gateway based on the interfaces available on the edge gateway.

Case 1 and 2 of the Data Center Cooling System, are evaluated based on the proposed characteristics of Collaborative computing paradigms and depicted in TABLES 6 and 7. It can be observed from TABLES 6 and 7 that Case 2 leverages the Collaborative Computing for achieving better functionality.

5.2 Vehicle and Driver Behaviour

In order to improve safety of the vehicles and drivers on the road, numerous solutions based on IoT have been deployed, which are based on the data collected from the On Board Diagnostics (OBD) port. Vehicle parameters like speed, engine rpm, odometer, coolant

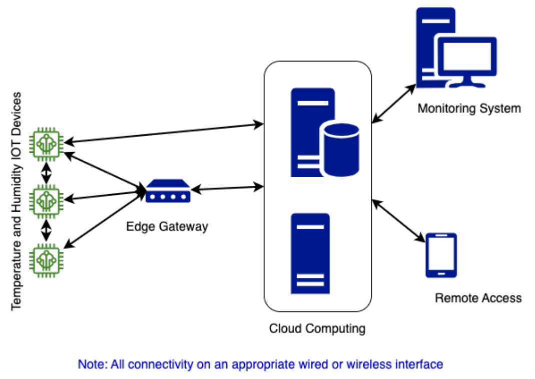
Temperature and Humidity IOT Devices
Edge Gateway
Cloud Computing
Monitoring System
Remote Access
Note: All connectivity on an appropriate wired or wireless interface

Figure 2: IoTD, CC, EC, MC: An architecture for Data Center Cooling System.

temperature, fuel level, fuel consumption, gear position are collected at frequent internals, from the Electronic Control Units (ECU) and are used for driver and vehicle behaviour analysis. Fault codes detected on the OBD can provide further information on need for service for the vehicle. Data is also collected from other sensors and smart devices like GPS, video from dashcam that provide vital information on location and environment. Based on these parameters further attributes can be computed like hard braking, acceleration, deceleration, adherence to speed limits which are useful for providing feedback to the drivers. Fault codes and other engine details can provide information on the running and fitness of the vehicle. Data collected from the vehicle can also be processed for predictive analytics for preventive maintenance.

5.2.1 Case (1) Vehicle OBD with IoT (Classical IoT and a Layered Approach)

An OBD device and a dashcam, installed on the vehicle, collects the data and transmits it to the Cloud for further processing. In some cases there may be a vehicle gateway that may collate the data, preprocess it - typically add time stamps, create packets of relevant corelatable data, example add location information, and organise it in files - before the data is sent to the cloud. Cloud server processes the data and provides the feedback in terms of alerts like excess speed, hard braking and acceleration to the driver and those are shown on the vehicle gateway or on the mobile phone of the driver. A mobile app, for the driver, provides the necessary processed information (received from the cloud server) in the form of information like driver score or alerts like hard-braking or acceleration.

In this case the complete processing done by the cloud servers, in event of network connectivity issues, the processed information and alerts will not be available for the driver. Additional features may be built on the mobile phone app which can perform minimal

Table 6: Collaborative computing for Data Center Cooling System.

Collaborative Computing Characteristics	Case 1	Case 2
Interconnection and interplay	NO	YES
Dynamic distribution of data processing	NO	YES
Fluidity of computing across paradigms	NO	YES
Storage and data management across participating paradigms	NO	YES
Scalability and extendability of the architecture	NO	YES

Table 7: Application: Data Center Cooling System.

Design of cooling system	Temperature Control	Electricity Optimisation	Scalability
Without IoT	YES. Localised	None	None
With IoT	YES. Cloud Controlled	YES. Cloud Controlled	YES. Need more computing on Cloud
IoT with Collaborative	YES. Edge Controller/Cloud guided	YES. Edge Controller/Cloud guided	YES. Cloud resources optimised

processing to alert the driver. Example: App on mobile phone may be able to generate hard braking or acceleration events to alert the driver.

5.2.2 Case (2) Vehicle OBD with IoT (Collaborative Computing)

In addition to the set-up as mentioned in Case (1), a vehicle gateway with better computing capability (Edge computing) is installed. The vehicle gateway typically will be equipped with multiple interfaces to connect with devices on board like the OBD, dash-cam and interface like Bluetooth or USB to connect to other devices like Mobile Phones or Laptop. The vehicle gateway can communicate with the Cloud server over a wireless connection. As in Case (1), the mobile phone of the driver can also participate in the overall solution.

With such an architecture, the vehicle gateway, the mobile phone and Cloud server can collaborate for preprocessing and processing the data. Example the hard braking, acceleration alerts can be generated on the mobile phone or the vehicle gateway, for speeding alerts the vehicle gateway can connect with the Cloud server to ascertain the speed for the particular road segment and compare with the actual speed of the vehicle, a collaborative method. Since the connectivity exists among all participating computing paradigms, depending on the load they can dynamically share the computing resources.

With all the computing paradigms (CC, EC and MC) working in a collaborative way, specific computing load can be shared with the EC or MC, thus increasing the overall capacity of the computing infrastructure. Figure 3 depicts the proposed Collaborative software system architecture.

Case 1 and 2 of Vehicle and Driver Behaviour, are evaluated based on the proposed characteristics of Collaborative computing paradigm and depicted in TABLE 8. As noted in description of earlier application, Collaborative Computing provides better functionality for identified characteristics.

Table 8: Collaborative computing for Vehicle and Driver Behaviour.

Collaborative Characteristics	Case 1	Case 2
Interconnection and interplay	PARTIAL	YES
Dynamic distribution of data processing	NO	YES
Fluidity of computing across paradigms	NO	YES
Storage and data management across participating paradigms	PARTIAL	YES
Scalability and extendability of the architecture	PARTIAL	YES

6 CONCLUSION

Based on the challenges and issues identified for computing paradigms and IoT, a collaborative computing system architecture is developed. The characteristics for the developed architecture have been identified and are mapped to the challenges and issues identified. It is clear from the mapping that the architecture developed is capable of addressing the challenges and issues. The proposed architecture is validated through it's application for vehicle and driver behaviour, and data center cooling systems. A comparison has been done for with and without the collaborative comput-

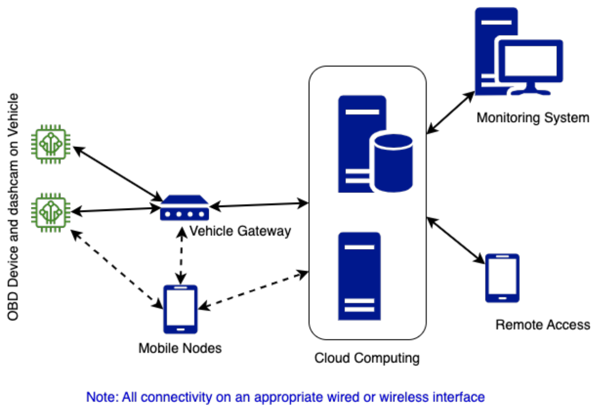
OBD Device and dashcam on Vehicle
Vehicle Gateway
Mobile Nodes
Cloud Computing
Monitoring System
Remote Access
Note: All connectivity on an appropriate wired or wireless interface

Figure 3: IoTD, FC/EC, MC, CC: An architecture for Vehicle and Driver behaviour.

ing architecture, For both the applications it is evident that this method adds dynamism, system reliability and higher utilisation of resources. It has demonstrated better capability to handle dynamic environment like IoT and better capability to handle uncertainty of connectivity issues. This architecture enhances the capability of computing paradigms characteristics when multiple paradigms are employed for a real-world system. With the capability to execute the tasks at any of the available computing paradigms or across multiple paradigms in a collaborative way, the overall higher system efficiency and utilisation of resources can be achieved.

7 FUTURE WORK

Further to this, the need is to experiment and apply for a variety of applications by focusing on the characteristics of interplay, dynamism, and fluidity of computing. With the collaborative method, the data storage across paradigms of preprocessed and processed data will need specific algorithms to collect the data at one location say Cloud. With the collaborative method, the distribution mechanisms of tasks across paradigms will need to be further studied along with effectiveness of AI and ML on dynamic environment like IoT.

REFERENCES

Baresi, L. and et al. (2019). A unified model for the mobile-edge-cloud continuum. In *ACM Transaction on Internet Technology*. ACM.

Bonomi, F. and et al (2012). Fog computing and its role in the internet of things. In *IEEE*. IEEE.

Cai, Q. and et al (2023). Collaboration of heterogeneous edge computing paradigms: How to fill the gap between theory and practice. In *IEEE Wireless Communications*. IEEE.

Carla, M. and et al. (2019). A comprehensive survey on fog computing: State-of-the-art and research challenges. In *IEEE Communications Surveys & Tutorials*. IEEE.

Donno, M. D. and et al. (2019). Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. In *IEEE Access, vol. 7, pp. 150936-150948*. IEEE.

Escobar, L. and et al (2022). In-depth analysis and open challenges of mist computing. In *Journal of Cloud Computing*. Springeropen.

Fernández, C. M. and et al (2018). An edge computing architecture in the internet of things. In *IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE.

Garcés, L. and et al (2021). Three decades of software reference architectures: A systematic mapping study. In *Journal of Systems and Software*. Elsevier.

Giovanni, M. and et al. (2019). Enabling workload engineering in edge, fog and cloud computing through openstack-based middleware. In *ACM Transaction on Internet Technology*. ACM.

Iorga, M. and et al. (2018). Fog computing conceptual model. In *NIST Standards and Technology*. NIST.

Kaushik, K. and Naik, V. (2023). Making ductless-split cooling systems energy efficient using iot. In *International Conference on Communication Systems*. COMSNETS.

Lewandowski, T. and et al (2020). A software architecture to enable self-organizing, collaborative iot ressource networks. In *Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE.

Mahmoudi, C. and et al. (2018). Formal definition of edge computing: An emphasis on mobile cloud and iot composition. In *Third International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE.

Malekian, R. and et al. (2015). Design and implementation of a wireless obd ii fleet management system. In *IEEE Sensors Journal, vol. 17, no. 4, pp. 1154-1164*. IEEE.

Mohammad, A. and et al (2018). Fog computing architecture, evaluation, and future research directions. In *IEEE Communications Magazine*. IEEE.

Munoz, R. and et al (2018). Integration of iot, transport sdn, and edge/cloud computing for dynamic distribution of iot analytics and efficient use of network resources. In *IEEE Journal of Lightwave Technology*. IEEE.

Nascimento, M. G. D. and et al (2023). An architecture to support the development of collaborative systems in iot context. In *IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE.

Peter, M. and Timothy, G. (2011). The nist definition of cloud computing. In *NIST Standards and Technology*. NIST.

Türker, G. F. and et al (2016). Survey of smartphone applications based on obd-ii for intelligent transportation systems. In *International Journal of Engg Research and Applications*. IJERA.

Vasconcelos., D. R. and et al. (2019). Cloud, fog, or mist in iot? that is the question. In *ACM Transaction on Internet Technology*. ACM.