Postprint of: Ribes V.S., Mora H., Sobecki A., Mora Gimeno F.J., Mobile Cloud computing architecture for massively parallelizable geometric computation, Computers in Industry, Vol. 123 (2020), 103336, DOI: 10.1016/j.compind.2020.103336

Mobile Cloud Computing Architecture for Massively Parallelizable Geometric Computation

Víctor Sánchez Ribes

PhD Student, Computer Science Technology and Computation, University of Alicante, Spain

e-mail: vsr37@alu.ua.es

Higinio Mora

PhD, Computer Science Technology and Computation, University of Alicante, Spain

e-mail: hmora@ua.es

Andrzej Sobecki

PhD, Dept. of Computer Science, Faculty of Electronics, Telecommunications and Informatics Gdansk University of Technology, Poland

email: andrzej.sobecki@pg.edu.pl

Francisco José Mora Gimeno

PhD, Computer Science Technology and Computation, University of Alicante, Spain

e-mail: fjmora@dtic.ua.es

***Keywords*.-**

GPU Computing; GPU; CUDA; Cloud offloading; Cloud Computing

Abstract.-

Cloud Computing is one of the most disruptive technologies of this century. This technology has been widely adopted in many areas of the society. In the field of manufacturing industry, it can be used to provide advantages in the execution of the complex geometric computation algorithms involved on CAD/CAM processes. The idea proposed in this research consists in outsourcing part of the load to be computed in the client machines to the cloud through the Mobile Cloud Computing paradigm. This practice gives substantial benefits to both the clients and the software-provider in terms of costs, flexibility, ubiquity and performance. In this document, an outsourcing architecture is proposed based on this paradigm. Extensive experiments have been done using highly parallelizable computational geometry operations to show the strengths and weaknesses of the proposal in combination of specialized computing platforms in the cloud. The results suggest that there are some issues that affect the overall performance and the stability of the QoS: the network communication delay, and the number of simultaneous clients and multiple requests. Some solutions have been proposed to face these challenges.

Corresponding author:

Dr. Higinio Mora

Telephone number: + 34 96 590 3400

Fax number: +34 96 590 3464

Postal address:

Department of Computer Science Technology and Computation

University of Alicante

Campus of San Vicente del Raspeig

03690 - San Vicente del Raspeig

Alicante, Spain

Acknowledgements

This work was supported by the Spanish Research Agency (AEI) and the European Regional Development Fund (ERDF) under project CloudDriver4Industry TIN2017-89266-R, and by the Conselleria of Innovation, Universities, Science and Digital Society of the Community of Valencia, Spain, within the program of support for research under project AICO/2020/206.

1. Introduction.

The new disruptive technologies such as Cloud Computing brings new advantages to the technologic infrastructure of the companies, including the CAD/CAM process. This paradigm constitutes one of the most innovative strategies regarding the adoption of Information Technology by companies. The advantages it provides allow for improved efficiency and reduced costs, while delivering ubiquitous accessible resources and services.

The manufacturing industry is also involved by this paradigm change. The use of the Cloud Computing in the industry provides a series of advantages for its design processes, such as the dynamic provisioning of resources by scaling services and provides almost immediate access to hardware resources, therefore the conversion of an adaptive infrastructure that can be shared by different users.

Among the many ways of applying the Cloud Computing paradigm to the manufacturing industry, this work proposes using Mobile Cloud Computing (MCC) to enhance the processing power of the software and to improve the computing services to manufacturer users. MCC is defined as the use of data processing and storage in the Cloud to reduce the workload on client devices, such as workstation, laptops, smartphones or tablets [1, 2]. Initially it was proposed to reduce latency and energy consumption of mobile devices, but over time, this technique has extended to other computing platforms [3]. Code Offloading is the method used to externalize the processing load to the Cloud [4]. This load can be an entire program, a function, a method, a process, or other piece of code. The Cloud computes this code and returns the result to the client. There are a lot of research developing and extending this technique to many scopes [2, 5]. Usually, the computing service is delivered though a Software-as-a-Service model.

Traditional manufacturing industry sectors (footwear, furniture, toys among others) are mainly characterized by having an intensive design and manufacturing work in producing new seasonal products. This work is made through 3D modelling and manufacturing software [6]. This software is usually known as CAD/CAM (Computer-Aided Design/ Computer-Aided Manufacturing). It is principally based on the application of geometric modelling and calculation primitives. An example of the use of this type of software is found in the modelling of objects and products, for example, in footwear industry [7], in the manufacture of children's toys [8] and in furniture designs [9], among others.

The application of this technique in the industry provides many advantages to the design and manufacturing processes: reduction of the initial cost for small and medium-sized companies that need high computing capacity, highly flexible infrastructure to provide adjustable computing power, delivering of CAD/CAM computing services to designers across the world, etc [10]. However, outsourcing geometric computation to the Cloud involves several challenges that need to be overcame in order to be feasible the proposal. These challenges include, among others, security and privacy in communications, applications, and data; connectivity and access to both the electric network as Internet must be considered as well; Cloud computing services must be reliable and able to support 24/7 operations, have contingency and tolerance plans; and the CAD/CAM applications must be prepared for code outsourcing [11].

This research deals with the study of outsourcing possibilities of CAD/CAM specialized software for the manufacturing industry that requires an intensive computational geometry calculation. This type of computations is characterized by being highly parallelizable. Thus, massively parallelizable devices can be used in the Cloud to speed up the processing load. This proposal provides additional advantages to the software users and medium and little manufactures, such as reducing the hardware

requirements at customer premises and enabling the use of mobile devices for complex calculations without additional hardware requirements.

In order to meet this goal, it is necessary to solve the following challenges: adaptation of the geometric calculation primitives, outsourcing of the specialized code for its processing in the cloud, the management of cloud processing for the execution of the primitives in specialized devices such as GPU (Graphic Processing Unit) and finally, the dimensioning of the operations to be carried out in an outsourced way [12].

The main novelty of this work is the definition of an application model of MCC technique for enhancing the geometric operations by computing them on the Cloud. Exhaustive experiments have been performed to prove the benefits of the proposal.

The remainder of the article is structured so that in a first point the existing developments in the areas related to this research are analysed. Later, the model is described through the formal definition of the problem and a conceptual view of the solution is provided. Point 3 describes the implementation of the model by introducing a technique. Point 4 presents a case study where the execution of an operation is proposed where the presented model is adequate. Finally, relevant conclusions and different future research lines are shown.

2. Related Work.

The following subsections discuss the state-of-the-art of the aspects related to this research. These are very intensive research areas, therefore, only the most representative and recent works are reviewed. A final subsection is added, which outlines the contributions to previous work.

2.1. Cloud Offloading

Cloud Computing is a model to allow ubiquitous and on-demand network access to a shared group of configurable computing resources that are easily accessed through communication networks [13].

Cloud computing can also be used to extend the limited computational resources of constrained clients by leveraging the resources and services of remote cloud [3, 14, 15]. To offload the processing work to the Cloud, you must decide how to offload. Either through static or dynamic offloading models. In both, there is the possibility of a partial offload of the processing or a full offload. Partial processing offload is based on the migration to the cloud of the part of the application that needs more processing power. The total of the processing offload involves migrating the entire application's processing to the cloud. All these techniques must be evaluated according to user and devices requirements such as delay, bandwidth or power consumption [16]. There is also the possibility of using a combination of different static type or dynamic type algorithms to obtain a greater benefit in the task of outsourcing processing [17].

Outsourcing methods that can be grouped into three large blocks according to their methodology: Client-Server, Virtualization and Mobile Agent methods [11].

The investigations grouped inside the Client-Server methods are mainly based on three communication protocols: Remote Procedure Calls (RPC), Remote Method Invocation (RMI) and sockets. There are many works and proposals of systems and architectures in this group. Next, a representative sample is described.

Spectra [18] and Chroma [19] are two examples of systems that use RPC for communication with the server. Hyrax [20] uses Hadoop 9 to enable the use of a group of smartphones as resource providers in addition to using RPC to indicate to the server

that they are alive. In addition, the work conducted by Huerta and Lee [21] presents a virtual mobile cloud, delegating tasks to other devices from an administrator, which is in charge of communication and the creation of virtual machines to execute the assigned tasks. Cuckoo [22] uses remote execution to execute and offload intensive calculation methods to the server. In order to improve performance and reduce battery, the works conducted by Deboosere et al [23] propose the use of a grid model together with a protocol such as VNC (Virtual Network Computing) to send the user inputs to the server returns the graphic output to the client mobile device. "MMPI" [24] is the mobile adaptation of the MPI standard, message passing, where Bluetooth technology is used for communication.

Regarding the migration of virtual machines or virtualization, there are also many developments such as: “Cloudlet” [25] where a framework based on the migration of virtual machines is proposed to outsource the calculation of the mobile device to nearby servers to avoid long transfer latency to the Cloud. "CloneCloud" [26] is a code outsourcing framework powered by "DalvikVM" [27] that implements one or more clones of the application and its data in virtual machines in the cloud and nearby servers using process interception to execute parts the cloud application. In “MAUI” [28], annotated methods are evaluated using decision logic at run time to decide whether to download or not. "COMET" [29] uses the Java memory model and virtual machine synchronization to create distributed shared memory for code migration between machines, so that the offload occurs by synchronizing the heap information, the memory stacks, registration states and synthetic classes between virtual machines in the Cloud and devices. "MobiCloud" [30] turns each device into a service node, which acts as a service provider or intermediary, depending on its available resources, and is incorporated, into the cloud as a virtual machine image. Finally, "CloudNets" [31] proposes a technique to handle virtual machine migrations between clouds, "live", in the background, transparently to the user.

In the mobile agent category, "Scavenger" [32] uses the mobile agent approach to partition and distribute jobs across devices. The work by Thompson and Morris-King [33] propose a framework used for the detection of malware in “MANET” type networks [34] in military environments. The work by Anjum et al [35] propose a robust anti-attack load balancer between devices for modification in mobile agents. The work conducted by Alsboui et al [36] show the data processing capacity based on IOTA in a scalable way and through P2P networks. Zilong et al [37] propose a multichannel access solution for outsourcing in industry 4.0 using a deep multi-agent reinforcement scheme. Finally, Salkenov and Bagchi [38] propose a software model to perform autonomous monitoring and administration of remote systems in large-scale heterogeneous structures.

Table 1 summarizes the main contributions made by the above previous, the outsourcing methodology used, and the application, system, technique or framework proposed in each work.

Table 1. Recent contributions on cloud offloading

Work	Main Contribution
Outsourcing method: client-server	
Spectra Remote Exec. System [18]	Use of RPC for invocation on both local and external servers.
Chroma Remote Exec. System [19]	It uses extra resources in over-resourced environments to improve performance.

Hyrax Map-Reduce Framework [20]	Based on Hadoop, it allows the outsourcing of tasks in heterogeneous networks of smartphones and servers.
Virtual Cloud Comp. Provider [21]	Use of smartphones for the creation of a virtual computing platform in the Cloud.
Cuckoo Cloud Computing Offloading Framework [22]	Code generation for the same implementation as in the local execution but characterized for the use of multi-core processors.
Algorithm [23]	Selection of servers automatically according to the location of the mobile device.
Outsourcing method: virtualization	
Map-Reduce Framework [25]	Re-assembly of the MapReduce framework.
CloneCloud Algorithm [26]	Optimal offline algorithm and near-optimal online algorithm to reduce runtime taking into account load balancing of all devices.
MAUI Runtime System [28]	Fine grain outsourcing.
COMET Runtime System [29]	New synchronization primitive of the DalvikVM virtual machine.
MobiCloud Security Service Architecture [30]	New security architecture.
CloudNets Technique [31]	Live migration between different cloud servers.
Outsourcing method: mobile agents	
Scavenger Technique [32]	Using the mobile agent approach to partition and outsource processing.
Agent-based Modellling Fwk. [33]	Malware detection in ad hoc mobile networks.
Outsourcing technique [35]	Using the hash function for outsourcing processing using Aglets
Outsourcing Technique [36]	Using Tangle for Distributed Intelligence.
Outsourcing Technique [37]	Implementation of a MADRL architecture for inter-device cooperation.
Software Architecture [38]	Combination of mobile agent-based approaches and script-based technologies.

2.2. Specialized Geometric Computation

This section shows different proposals of specialized cloud processing. These works describe methods, techniques, frameworks and applications where specialized geometric calculus processing are offloaded to the Cloud.

The simplest case consists in using as computing platforms the CPU of the cloud server itself since it is usually most powerful than client's processors. However, specific computing platforms are increasingly used in the cloud to enhance the processing and take advantage of the features of massively parallelizable geometric calculation of the tasks. In this way, the cloud server can be equipped with GPUs and FPGAs devices to perform this kind of calculations. In this subsection, several examples of each type are reviewed.

In first place, the most common case is using the server CPU to make the calculations. In this set, the following works are highlighted: the work conducted by Sharma and Wang [39] proposes a framework for processing that uses the advantages of Cloud and Edge Computing to provide mechanisms for extracting useful features from huge amounts of heterogeneous data. Another computing framework is proposed by Ouahmane et al [40] to secure processing of digital images using segmentation and watermarking techniques. The work by Xia et al [41] propose the use of an effective outsourcing protocol ensuring privacy and Local Binary Pattern (LBP) functionality on large encrypted images. The "LightCom" system [42] allows data storage and processing on a single server by presenting two sets of tools for fast and secure processing of floating-point numbers. The work by Zang et al [43] proposes a protocol for outsourcing Eigen-decomposition and single-value decomposition for face recognition by means of Principal

Component Analysis (PCA). The works by Kalideen and Tugrul [44] outsource the calculations for the k-Nearest Neighbour (k-NN) to the cloud and returns the results to the device that has performed the query. Bao and Li [45] present a geometric programming externalization protocol encrypting the original problem in the client side and using the gradient projection method the problem is solved in the server. Furthermore, Danevičius et al [46] show the "Jelly Dude" game that uses outsourcing to ensure high quality graphics using the Jordan Neural Network (JNN). Silva et al [47] propose a framework for encrypting homomorphic images with which can be processed while they are encrypted. Li et al [48] propose a solution for secure geometric range queries by means of a multidimensional range general query and use the R-tree index to obtain efficiency in sublinear searches. Vo et al [49] propose a generic and highly scalable cloud-based framework for image analysis that allows the parallel integration of image analysis steps and the generation of database driven objects. The work by Lu et al [50] evaluates different techniques for reducing the size of control commands issued to a 3D printer from a Cloud-based controller (C-CNC) by achieving high quality prints at high speed from the controller in the Cloud through high-latency Internet connections. Finally, in the same line, Wang et al [51] and Okwudire [52] propose a ubiquitous and collaborative three-dimensional printing system, which reduces manufacturing time.

In second place, there are works that propose using the Cloud Computing infrastructure in combination of GPUs specialized processors in order to improve the performance [53]. Thus, this GPU-cloud computing combination increase both speed and accuracy of advanced applications such as facial detection systems [54], fluid simulator [55] or a structure calculation [56]. Other proposal that use the GPUs to optimize and improve the performance is GPUNFV [57], which proposes a GPU-accelerated NFV (Network Functions Virtualization) system.

Finally, other devices also play an important role in complementing Cloud Computing infrastructure. For example, the work by Lallet et al [58] proposes a FPGA system for concurrent acceleration of native Cloud microservices. The proposal by Ojika et al [59] present a high-performance architecture with FPGA as a microservice for the Cloud. Jiang et al [60] show the advantages of using FPGA for interactive applications at the network edge. Qingqing et al [61] present the use of processing outsourcing for odometry using FPGAs.

Table 2 summarizes the main contributions reviewed classified by the main cloud computing platform used to compute the tasks and the application, system, technique or framework proposed in the works.

Table 2. Recent contributions on specialized geometric calculations processing in the cloud

Work	Main Contribution
Cloud processor: CPU	
Framework for coordinating edge and cloud computing [39]	Identification and description of key enablers for edge-cloud collaboration.
Computing framework [40]	Use of segmentation and watermarking to ensure data privacy.
Outsourcing protocol [41]	Feature extraction from encrypted images using secure LBP extraction.
LightCom system [42]	Formalization of geometric encryption for search.
Outsourcing protocol [43]	Using PCA for facial recognition.
Computing framework [44]	Using the k-NN method with a kd-tree
Outsourcing protocol [45]	Using the gradient projection method to solve the encryption on the Cloud side.

Soft physics simulator [46]	Outsourcing of processing through the Jordan Neuronal Network.
Computing framework [47]	Encrypted image processing.
Cloud Geometric Query Tech. [48]	Re-focusing of the geometric range query in a cloud data set.
Cloud-Based CNC for a 3D printer [50]	Use a technique to reduce the size of the data for 3D printing from the Cloud.
Image analysis framework [49]	Abstraction of images at different types of spatial geometries.
Collaborative and ubiquitous system for 3D printer [51]	Creation of a cloud-based linear programming model of mixed integers and a quadratic programming model of mixed integers.
Cloud processor: GPU	
GPGPU Based Acceleration Offloading [53]	CUDA Virtualization and Remoting for GPGPU Based Acceleration Offloading
Mobile GPU Cloud Computing [54]	Mobile GPU Cloud Computing with real time application Facial detection system
General Purpose GPUs [55]	Virtualizing General Purpose GPUs for High Performance Cloud Computing
TeraChem Cloud [56]	High-Performance Computing Service for Scalable Distributed GPU-Accelerated Electronic Structure Calculations.
GPU-accelerated NFV system [57]	NFV (Network Function Virtualization) System Efficiency using GPUs and CUDA language.
Cloud processor: FPGA	
Cloud Architecture [58]	Share the same FPGA for offloading functions for different microservices.
Architecture as microservice [59]	Proposal of a FPGA Architecture (FaaM) for providing auto-deployment, scalability, dynamic configuration and disaster recorvery.
Network-assisted cmp. model [60]	Acceleration of Computer Vision applications.
Edge Computing technique [61]	Proposal of an odometry logarithm modelled with VHDL and accelerated with FPGAs.

2.3. Cloud Offloading for Geometric Calculation

This section reviews some works where the cloud has been used to compute geometric calculations. The scope of the applications is wide, and it includes mobile applications (Apps), intelligent vehicles, games and industry, among others. However, these methods are not widespread enough and the set of proposals is limited. There is a lack of research on adapting the existing algorithms to the offload methods in order to take full advantage of the cloud resources.

One of the fields where this technique has been widely used is in the mobile computation area. This aims to provide extra capabilities for smart devices such as mobile phones, smart watches and other wearables. As mentioned previously, these capabilities have traditionally consisted in increasing the hardware performance by minimizing the energy consumption and improving the battery life. Nevertheless, the potential of MCC paradigm allows to provide advanced calculation methods to these devices and other computations platforms as it is proposed in this work.

In the area of mobile computing, the work by Chung et al [62] proposes to generate Augmented Reality (AR) applications on smart devices by using adaptative MCC techniques to offload the AR work to the cloud. This AR work consist in producing virtual objects to be displayed on the mobile screens combined with the physical reality. These objects are realistic geometric designs that are placed on the scene. In this same line, other works propose to offload the AR work to other computing platforms [63].

The intelligent vehicles and upcoming autonomous vehicles are other application area where the offloading techniques has gained widespread popularity recently. In this field, the geometric calculations play a critical role in the performance of the system, for example, in providing driver-assisting AR tools [64] or in enabling real-time object detection to understand the 3D geometry of the surroundings [65].

Gaming is another growing area of GPU-CPU cloud combination. There are works which propose using GPUs in the Cloud to provide a high-performance Cloud Gaming System using frame rendering directly on GPU cards over the Cloud [66]. This system offers a "ShareRender" system to deliver computer games to users. The involved geometric calculations are computed on the cloud and delivered through Internet to gamers with thin clients on heterogeneous devices [67].

Finally, the industry field is open to new possibilities of advanced cloud computing. For example, the work by Lynn et al [68] proposes a general purpose virtualized computing platform (GPGPU) for the development of a voxelized CAM package. Voxels are essentially 3D pixels, but instead of being squares, they are perfect cubes. Therefore, complex geometric calculations are needed to build complex objects. Other works show the use of Computer Aided Engineering (CAE) technology and the power of the Cloud to realize an integrated system of Cloud-based CAE simulation for industrial applications where complex calculations are involved [69].

Table 3. Recent contributions on specialized geometric calculus processing in the cloud

Work	Main Contribution
Augmented Reality Applications on Smart Devices [62]	Augmented Reality computations is performed on the cloud server by using adaptative MCC techniques.
ARVE [64]	Augmented Reality Applications in Vehicle to Edge Networks
Edge Assisted Real-time Object Detection [65]	System that enables high accuracy object detection to understand the 3D geometry of the surroundings.
Cloud gaming system [66]	Implementation of an online algorithm for the outsourcing of the processing as well as the migration of the rendering agents.
Cloud Gaming [67]	Fine-Grained Scheduling in Cloud Gaming on Heterogeneous CPU-GPU Clusters
General purpose virtualized computing [68]	CAM development using CUDA language package that generates toolpaths for highly complex computations.
Cloud-Based CAE simulation system [69]	CAE (Computer Aided Engineering) analysis system on Cloud servers

2.4. Findings

After reviewing the previous research, some conclusions can be drawn to justify the contributions made to the state of the art.

- There is high interest in research on cloud computing and cloud-processing offloading techniques to increase the processing power of devices with limited resources. In this way, computing power of client devices can be homogenized to run applications with high computing requirements transparently to the user.
- Cloud datacentres can be equipped with specialized computing platforms such as GPUs to increase the performance of computing high parallelizable algorithms when they are offloaded to the cloud.
- Geometric computation and other massively parallelizable methods involved on CAD/CAM software are good candidates to be processed under the Mobile Cloud

Computing paradigm in order to transfer these advantages to manufacturing industry and to software providers.

In line with these findings, proposing a distributed architecture for offloading the geometric computation methods to the cloud is one of the contributions of this work. In addition, in order to take full advantage of the current infrastructure possibilities, these methods should be optimized for GPU processing. The combination of these two aspects (Cloud offloading + GPU processing) are the key contributions and novelties of this research. Other secondary aspects of this proposal are also analysed such as the impact of the network delay on the overall performance and the QoS stability. The result enables the use of lightweight software with high computational power, which in turn allows to the customers use thin clients as smartphones or tablets to perform complex CAD/CAM operations without additional hardware requirements.

This operation mode means a significant change in the business model of the CAD/CAM software providers, who need to make an effort in adapting their software and build the required Cloud hardware and management methods.

3. Distributed geometric computation model.

In this section, a computational model for specialized geometric computation is proposed. This model is based on the foundations of distributed computing under the paradigm of mobile cloud computing.

This proposal has three parts that may be analysed: (i) the processing of the application itself, (ii) the cloud architecture, and (iii) the communication network.

(i) In first place, the geometric application is run on the client platforms, and then, to externalize a piece of software to be processed in the cloud when needed. Some aspects must be taken into account to achieve this behaviour.

The most important is that the application must be ready to be outsourced without penalize the user experience. In this case, geometric applications are characterized by being massively parallel due to the nature of the data they work. This data usually represents 2D and/or 3D images or figures, and it composed by vectors, arrays, pixels, polygonal figures, frames, etc. without hard dependent relationships among them. Thus, it is easy to prepare an isolated piece of code to processing part of the data, for example, and to externalize it to be compute in the cloud without affecting the operation of any other part of the application. Indeed, most of the CAD/CAM software is already prepared to be computed on a parallel way in order to take advantage of the parallel computing platforms at hardware level, such as modern GPUs and DSPs. At present, it is the most common case of computing the CAD/CAM applications. The software is computed on premise machines equipped with advanced GPUs devices. This situation causes a high entrance-cost of clients, who need invest in expensive hardware in order to run it properly.

The user must configurate the system to setup the client-device according the needs and the contract with the software provider. Here, some business models can be applied: pay-per-license model (traditional), Cloud Computing model, and a mixed model according to the specific necessities of each scenario. The outsourcing method to perform the calculations could be any of the described in table 1.

(ii) Regarding to the cloud architecture, there are some challenges that must be addressed.

As known, the 2D/3D design software are heavy in floating-point operations and requires the execution of complex mathematical functions of geometrical computation. In addition, the increase the realism of 3D designs and graphics of recent times largely

exceed the capabilities of conventional general-purpose CPU platforms. This type of specialized software is usually computed at Graphic Processing Units.

However, the cloud-processing infrastructure is mostly composed by standard CPUs. Although some cloud providers already offer this specialized computing service, there is a challenge to meet the QoS requirements of advanced design-software. The cloud operating systems need to optimize the execution of operations by maximizing the hardware utilization and minimising the delay.

(iii) Finally, the communication network and the protocols involved. In the general case, this network operates via the whole Internet since the cloud provider could be located at any place of the world. However, other options for deploying the infrastructure could be considered with the aim to provide processing efforts as close as possible where the data is generated. This is the case of the cloudlet infrastructure. A Cloudlet is a cloud server of a lower scale. They are deployed within a geographical area where many potential clients are [70], and even form part of the same Local Area Network than clients. For example, a 3D design company could deploy a cloudlet to provide computing services to its employees' demands. Other multi-layer options can be designed to address more complex scenarios [71]. In all cases, Internet Protocol (IP) will be necessary to provide internetworking communications and move data across networking boundaries in a consistent manner.

Fig.1 shows a scheme of the overall distributed computation model proposed in this work where each part of the model described above is depicted.

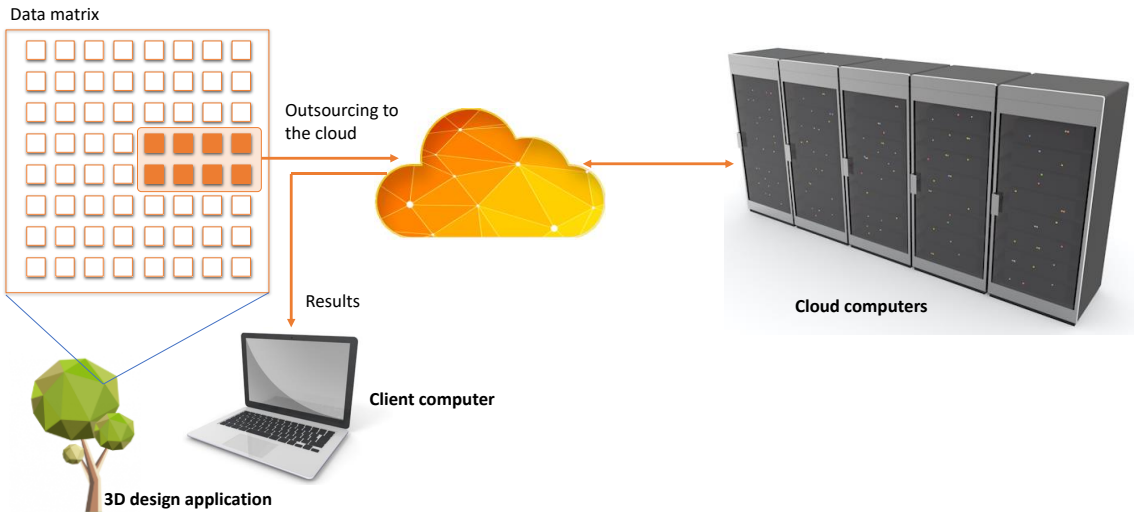

Fig. 1. Distributed geometric computation model

As a result of this computation model, an enriched business of design software delivery arises with many possibilities both for software providers and customers around the world. In this way, customers may continue to have powerful on premise, infrastructures to compute the software, also they can use light client-devices to run the same software (mobile devices, tablets, personal computers, etc.) where part of the processing load are outsourced to the cloud, or they can have devices that switch the operation mode depending on the remain battery power of the device, QoS (Quality of Service) requirements, available bandwidth, and monetary cost of the cloud services at each time, among others.

Apart from the described previous parts of the distributed architecture, other key components are needed for implementing the model in a real-world application: the client-scheduler and the cloud-scheduler.

The client-scheduler is part of the Operating System (OS) of the client computer, which must support this operation mode. It operates as a common OS scheduler considering offloading to the cloud as other available processing unit of the platform. Fig. 2 depicts this scenario.

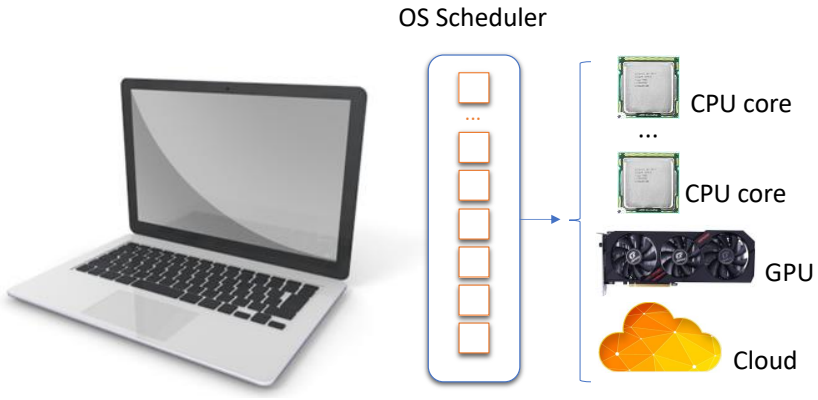

Fig. 2. Available processing units of the client computer

This component decides for each task where is processed among the available computing elements. In traditional devices, it mainly depends on the nature of the task. That is, a graphic-nature task is processed by the GPU (if present) while a standard task is processed by the next available core of the processor.

In this case, the client-scheduler needs to decide what, when and where to externalize the processing load without user intervention. Although, in first place this practice was used in mobile computing to minimize the energy consumption and improve the battery life [72], now, several criteria can be considered to decide offloading. For example: computing time, available bandwidth or monetary cost of the cloud services, among others [71].

In the application case described in this work, the computing delay is the most relevant feature. This computing time (T) has the following components: (i) Client-Processing time (T_{Client}): this is the execution time of the task in the client computer, (ii) Cloud-Processing time (T_{Cloud}): this is the execution time of the task in the cloud server, and Communication time (T_{Comm}): this is the delay added by the network. In addition, the communication time consists of the time of sending the data to the server (T_{Send}) and the receiving the results from it (T_{Receiv}). The following expression compiles all the components in the same formula:

$$T = \begin{cases} T_{Client} & if\ the\ task\ is\ processed\ in\ the\ client\ computer \\ T_{Send} + T_{Cloud} + T_{Receiv} & if\ the\ task\ is\ processed\ in\ the\ Cloud \end{cases} \quad (1)$$

All these time components may vary over time depending on the processing and communication conditions. Thus, all are a function of time.

The Client-Processing time (T_{Client}) is known and predictable, that is, the scheduler knows its value at any time. However, the other components may be estimated in order to make a scheduling decision.

The Cloud-Processing time (T_{Cloud}) should be also known and stable; however, it depends on the computing load of the server. In a normal situation, the cloud provider shares the infrastructure with many clients. Thus, it should adapt their servers to the real demand at any time. Some techniques are needed to scale and adjust the available hardware to the requirements of each time (load balancers, switch on/off servers and specialized cards) in order to keep the time delay within a reasonable range. The following figure shows a typical configuration of the server infrastructure.

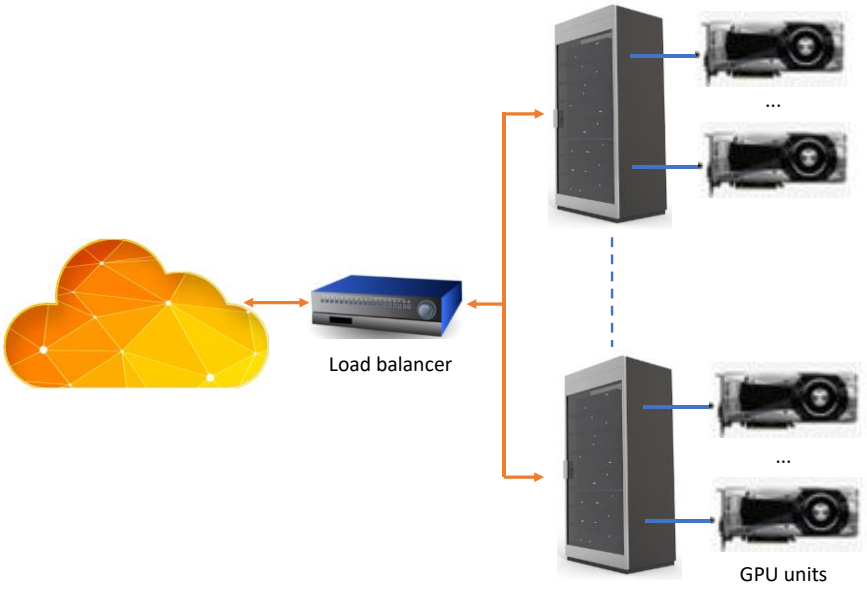

Fig. 3. Common specialized Cloud infrastructure

As shown, the cloud infrastructure could add load balancers to properly distribute the tasks and absorb the dynamic loads. In addition, each cloud server can be equipped with several GPU units. In this way, the cloud provider could adapt their infrastructure to the number of potential clients. Although they might have a predictable behaviour, some peaks of demands could arrive. In these cases, the time delay can be affected.

Finally, the Communication time (T_{Comm}) is the most variable aspect of this architecture because it is susceptible to unexpected changes due to fluctuations in bandwidth and response delay that may affect the QoS. The stability of a shared medium network depends on many factors such as the number of connected users to Internet, the applications demands, presence of external interferences or noises, the geographical location, etc.

There are some techniques to address this issue and forecast the network reliability and availability. For example, the client platforms can analyse the network quality to know the performance at each moment. In addition, predictive analysis based on machine learning algorithms that uses historical data on the performance of the network can be also applied to get a smooth prediction [15, 73]. However, this is still an interesting open issue in the internet era with great impact on the deployment of connected applications. Hopefully, the recent technological advances in communication technologies, as 5G wireless networks, play in our favour to provide stability to the network performance. It is expected that the upcoming 5G will bring near-to-zero latency and “five 9s” availability [74]. These capabilities will be a pivotal enabler of emerging usage scenarios and applications such as the proposal in this research.

Taking into account the previous considerations, the client-scheduler must decide to offload in the following cases: when the user has configured this operation mode, when the client computer cannot process the tasks and when the time delay of the task processed in the Cloud is lower than processed in the client computer. That is represented in the following expression:

$$T_{Client}(\text{time t}) > T_{Send}(\text{time t}) + T_{Cloud}(\text{time t}) + T_{Receiv}(\text{time t}) + \delta \quad (2)$$

where t is the time instant where the decision is taken, and δ is a threshold to avoid fluctuations and unforeseen delays.

In the other side, the cloud scheduler is in charge of decide in which node of the cloud infrastructure will be execute each incoming task. The main purpose of scheduling is to produce an execution plan, which allows completing all tasks of a workflow within given time and cost constraints [75]. However, at this stage, it is not easy scheduling the offloaded tasks into the CPUs core or into a GPUs node. There are some critical challenges such as the overheads the cloud providers introduce, data access and parallelism limits of the GPUs. In addition, there are a heterogeneity in the tasks and in the computing resources.

In the next section, an extensive experimentation is conducted to validate the proposal, to determine the timing framework and to increase our knowledge about how this paradigm works. To this end, several experimental scenarios have been configured.

The results obtained will allow us to advance in the proposal of an architecture to compute in the cloud CAD/CAM applications, and thus, be able to exploit the full potential of this disruptive technology in the manufacturing industry.

4. Experiments and results.

Once the proposed model for the outsourcing of geometric processing has been presented, a set of experiments has been designed as proof of the viability of the proposed model.

4.1 Experimental Configuration

The proposed architecture has many elements to configurate and decisions that need to be made. A fine calibration of these elements is very important to get a proper operation of the model. This section describes some key choices of the architecture and the description of the computing platforms involved.

The outsourcing method used in the experiments is based on client-server model, and the communication protocol used is TCP/IP due to its wide use, popularity in industry and ability to ensure the communication between components. It also serves as a communication channel for different current developments [76, 77]. With this protocol, the client computer will make a calculation request to the server through the TCP socket, where it will send all the necessary information for the calculation and its later return to the client computer.

The computing platforms have the following configurations described in Table 4.

Table 4. Computing platforms

	CPU	RAM	GPU
Client platform	i7-7700HQ+HM175 (3.80 GHz)	8 GB	NVIDIA GeForce GTX 1050, 2GB 640 CUDA cores
Cloud platform [A]	Intel Pentium G850 (2.90 GHz)	8 GB	NVIDIA GeForce GTX 480, 1.5GB 480 CUDA Cores
Cloud platform [B]	Intel Xeon E5-2690 v3 (2.60 GHz)	56 GB	NVIDIA Tesla K80 4.992 CUDA Cores

[A] Cloud platform deployed at the Computing Technology Department of the University of Alicante, Spain.

[B] Cloud virtual machine contracted to Azure

The experimentation carried out in this section is based on the execution of a vector sum of floating-point elements in different platforms and execution conditions. This simple operation is involved on many geometric and graphic transformations. In addition, this operation can be highly parallelised using parallel programming techniques. The parallel version of this operation has been implemented in NVIDIA's CUDA (Compute Unified Device Architecture) which provides the benefits of parallelism obtained by using multi cores GPU devices (Fig. 4).

```
extern "C"
__global__ void add(int n, double *a, double *b, double *sum)
{
   int i = blockIdx.x * blockDim.x + threadIdx.x;
   if (i<n)
   {
      sum[i] = a[i] + b[i];
   }

}
```

Fig. 4. CUDA kernel code

4.2. Experiments

At this point, a series of experiments have been carried out in order to find the degree of affectation of the most suitable communication system for the construction of this model.

Experiment 1

For this experiment, the vector floating-point sum code depicted in Fig 4 has been processed by the three platforms described in Table 3 for different vector sizes. The objective of this experiment is to show the performance gain of using specialized computing platforms. The results (Fig. 5) show the performance of the different platforms used.

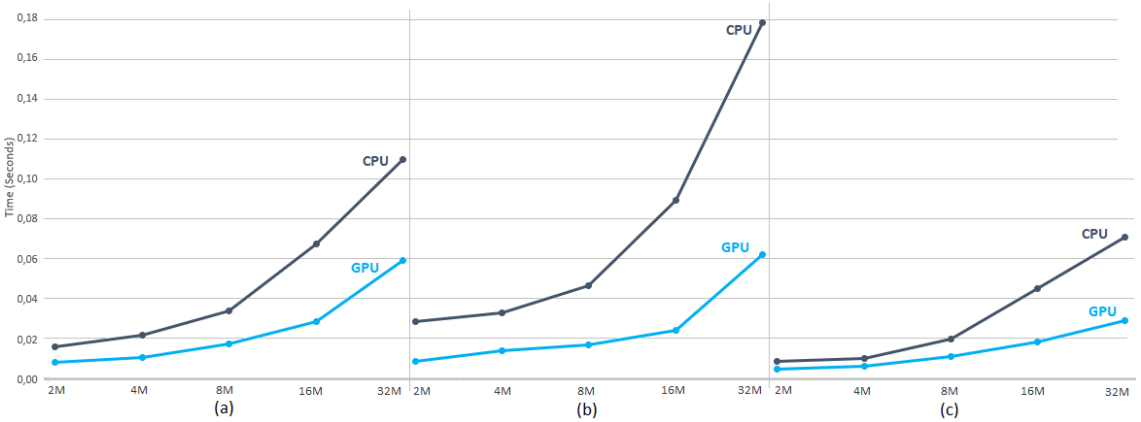

Fig. 5. Serie vs parallel computation. (a) Client Platform; (b) Cloud A platform; (c) Cloud B platform

The results of this experiment clearly illustrate the better performance of the GPU devices in computing massively parallelizable operations. Other features such as clock frequency or number or cores also influences the time processing.

Experiment 2

In this experiment the communication costs of outsourcing the workload to the cloud are obtained. This communication costs are composed by the cost of sending the load (T_{Send}) and receiving the calculated results (T_{Receiv}), as shows the next expression:

$$T_{Comm} = T_{Send} + T_{Receiv} \quad (3)$$

These costs depend on the volume of data and the communication technology used. Thus, several experiments have been conducted using different data sizes and communication technologies. Fig 6 shows T_{Comm} for different communication technologies LAN, Wi-Fi and 3G.

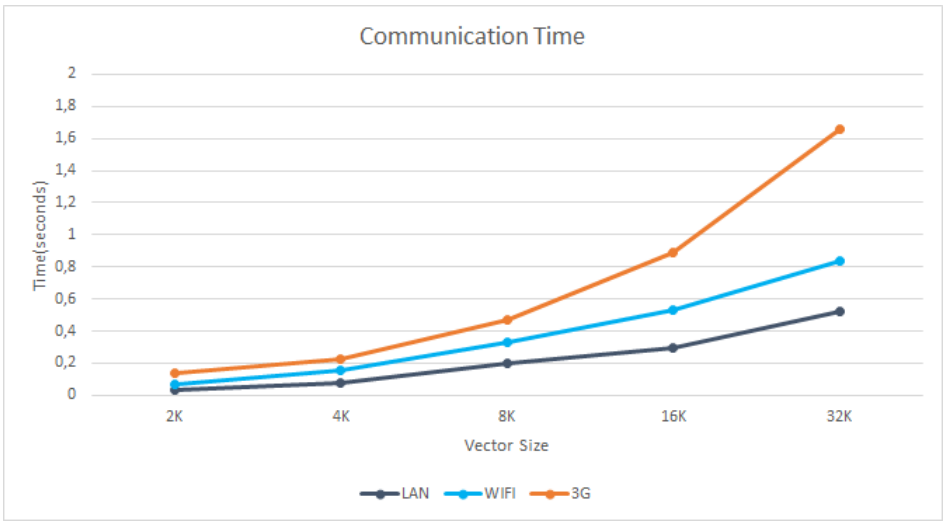

Fig 6. Communication time with different communication technologies.

The results show how the quality of the network impacts in the final cost of communication.

Experiment 3

In this experiment, a comparison between local and remote execution is performed taking into account all components of the costs described in expression (2). Several scenarios have been designed in order to describe different possibilities.

Fig 7. shows the results between *Client Platform* and *Cloud Platform B* GPU execution.

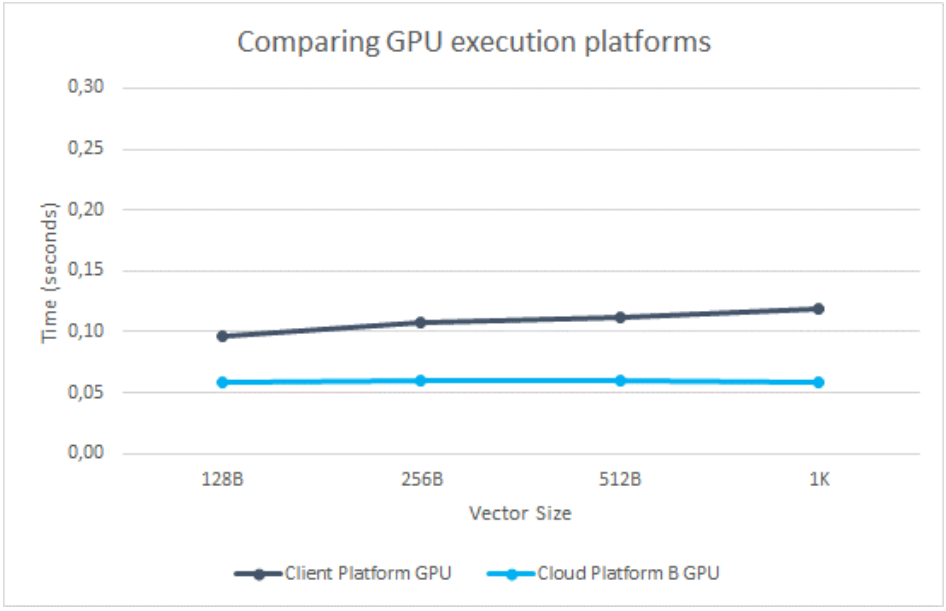

Fig 7. Comparing executions between platforms with GPU.

Fig 8. shows the time costs between *Client Platform* and *Cloud Platform B* CPU execution.

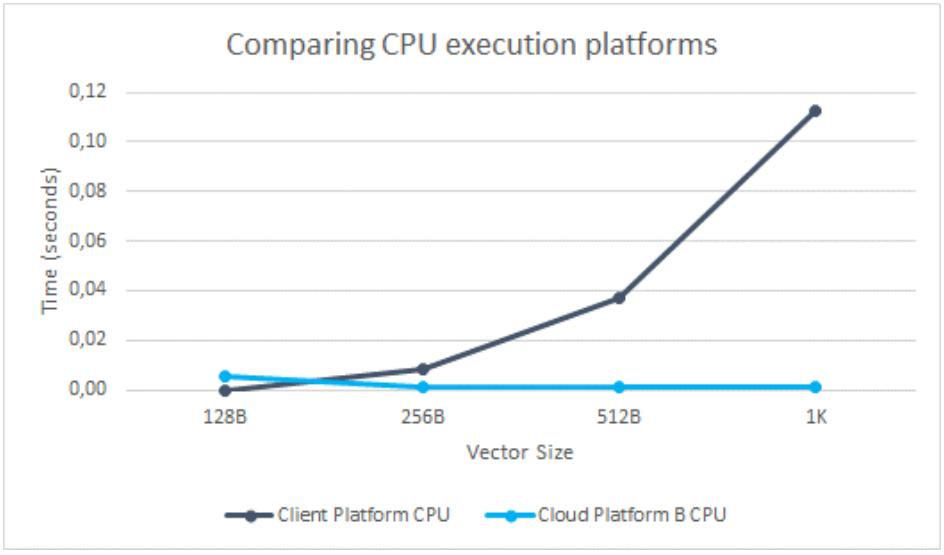

Fig 8. Comparing executions between platforms without GPU.

In order to reduce the amount of data to be transmitted each time, and therefore the communication costs, the whole working data set could be preloaded previously on

the server, and only to send the operation codes in the offloading operations to command the remote software what to do with the pre-loaded data.

The operations performed by the CAD/CAM software are complex mathematically which involves high computational costs. This kind of operations on images and 3D models can be massively parallelized for achieving greater performance and to leverage the features of modern GPUs., shown in Fig. 5. The results obtained are of great interests for industry since the client machines around the world do not need to have great performance to work with complex CAD/CAM software. According to the outsourcing policies, the customers of this software can use it under SaaS business model and make the most of all advantages of cloud computing paradigm.

For example, in the case of our example, the GPU of Client-platform (depicted in table 3) costs about $400 and consumes 75W of power. A median organization with around 50 design workers, can save $20,000 in equipment and up to 75Kw of power/month by using SaaS design software.

Experiment 4

In this experiment, a dynamic load test has been designed in *Cloud Platform B*. In this experiment, a massive request has been made to the cloud server in order to observe the behaviour in the response time. For this purpose, threads have been configured to simulate the simultaneous request to the platform (Fig. 9).

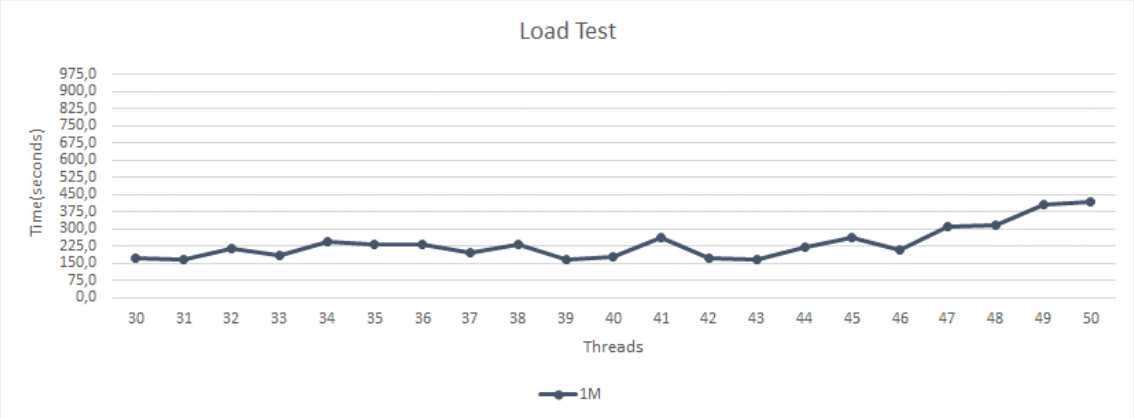

Fig 9. Load Test results with 50 clients.

This experiment shows that simultaneous requests maintain the stability of the system up to the 47th and 48th request using 32K vectors. A single server with only a GPU has been used. Of course, a production cloud architecture should forecast the future demand in order to provide enough the computing power and maintain the delay within the expected QoS.

Experiment 5

This experiment is focused on validating the model with a computational geometry function. The function is the normal vector calculation of each point of a 3D mesh. This function is essential for the implementation of modern CAD/CAM software for 3D design. Next figure shows an example of 3D mesh and the normal vector of some points (Fig. 10).

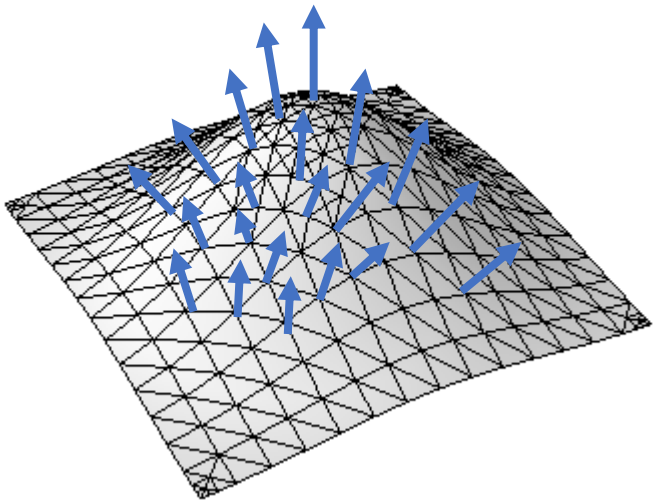

Fig. 10. Normal vectors of each point of 3D mesh

Two sets of experiments have been conducted over the same architecture described in this work.

The first experiment shows the processing time comparison for serie vs parallel computation in the Cloud (Fig. 11).

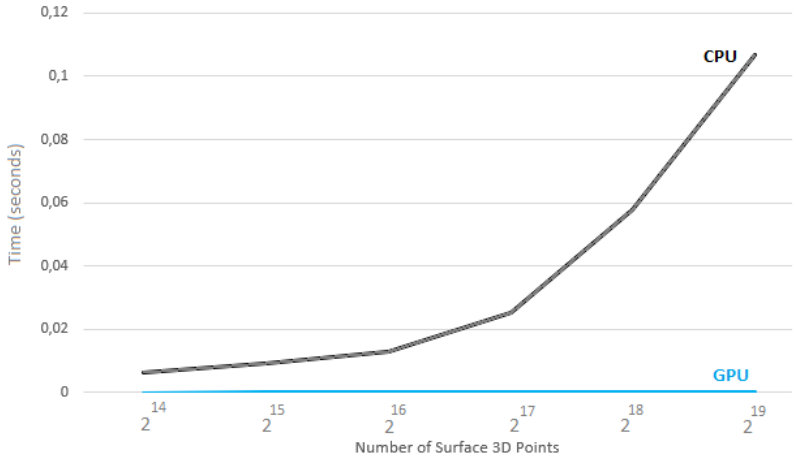

Fig. 11. Serie vs parallel computation in Cloud B platform

In this case, the geometric function is highly parallelizable. Thus, the gain obtained by advanced GPU platforms in the Cloud can provide high performance computing for CAD/CAM applications.

The second experiment compares the performance between Client Platform and Cloud Platform B GPU execution (Fig. 12). The Cloud processing takes into consideration all cost involved in the cloud computing as described by expression (2).

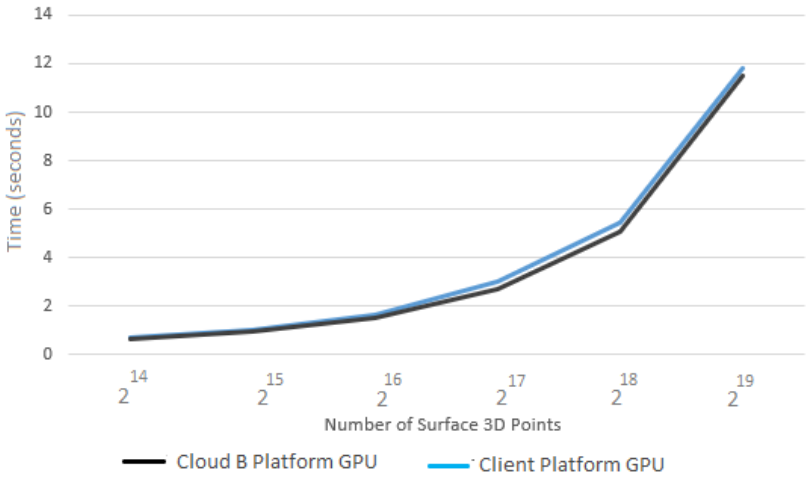

Fig. 12. Client vs Cloud Platform computation

As shown in the previous figure, the results obtained in this experiment demonstrate that for certain massively parallelizable functions, the better performance obtained in the Cloud GPU platform compensates the communication cost to offload the work. Thus, it is worth to research new methods for leverage the cloud resources and to adapt the CAD/CAM software for the new computing paradigms and architectures in the cloud.

The results obtained are of great interests for industry since the client machines around the world do not need to have great performance to work with complex CAD/CAM software. According to the outsourcing policies, the customers of this software can use it under SaaS business model and make the most of all advantages of cloud computing paradigm.

5. Conclusions.

The present work shows a state of the art of the investigation on the main applications of Cloud Computing as well as of the CAD/CAM software. The findings show a lack of research in the application of this paradigm with GPU devices and the use of massive parallelization languages such as CUDA applied to specialized software for computer-aided design and manufacturing.

In this work, a distributed computing model has been described to explore the application of Mobile Cloud Computing paradigm to perform massively parallelizable software with specialized GPU devices in the Cloud. This model offers ubiquity in resources and services, and provides computing power without the need for physical dependence on expensive graphics cards highly consumers of power. Several experiments have been carried out to validate the proposal and show the benefits of using SaaS model for this industry.

The results also show the importance of the communication costs with the Cloud. These costs depend on the volume of data and on the communication technology used. This is a key factor to know the expected performance of the offloading method. However, the development and deployment of better communication technologies such as 4G, 5G and optical fibre will increase the transmission speeds through the network and, therefore, reduce the communication delay of this model.

For future research work, further effort must be invested in studying parallelizable software as CUDA and thread parallelization techniques by improving the core of the graphics card and local memory management. Other interesting research one is about new communication strategies to reduce the delay and minimize the latency.

6. References.

[1]. Biswas, M., & Whaiduzzaman, M. (2018). Efficient Mobile Cloud Computing through Computation Offloading. International Journal of Advancements in Technology, 10(1), 1–7. https://doi.org/10.4172/0976-4860.1000225

[2]. Nawrocki P., Reszelewski W., Resource usage optimization in mobile cloud computing, Computer Communications 99 (2017) 1–12, https://doi.org/10.1016/j.comcom.2016.12. 009

[3]. Mora, H., Colom, J. F., Gil, D., & Jimeno-Morenilla, A. (2017). Distributed computational model for shared processing on Cyber-Physical System environments. Computer Communications, 111, 68–83. https://doi.org/10.1016/j.comcom.2017.07.009

[4]. Mahmoodi, S. E., Uma, R. N., & Subbalakshmi, K. P. (2019). Optimal joint scheduling and cloud offloading for mobile applications. IEEE Transactions on Cloud Computing, 7(2), 301–313. https://doi.org/10.1109/TCC.2016.2560808

[5]. Khan M.A., A survey of computation offloading strategies for performance improvement of applications running on mobile devices, J. Netw. Comput. Appl. 56 (2015) 28–40, https://doi.org/10.1016/j.jnca.2015.05.018

[6]. Davia-Aracil M., Hinojo-Pérez JJ., Jimeno-Morenilla A., Mora-Mora H., (2018) 3D printing of functional anatomical insoles, Computers in Industry 95, 38-53. https://doi.org/10.1016/j.compind.2017.12.001

[7]. Orazi, L., Reggiani, B. (2020) Innovative method for rapid development of shoes and footwear. The International Journal of Advanced Manufacturing Technology 106, 2295–2303 (2020). https://doi.org/10.1007/s00170-019-04717-8

[8]. Mircheski, I., Lukaszewicz, A., Trochimczuk, R., & Szczebiot, R. (2019). Application of cax system for design and analysis of plastic parts manufactured by injection moulding. https://doi.org/10.22616/ERDev2019.18.N463

[9]. Sun, X., & Ji, X. (2020). Parametric Model for Kitchen Product Based on Cubic T-Bézier Curves with Symmetry. Symmetry, 12(4), 505. https://doi.org/10.3390/sym12040505

[10]. Avram, M. G. (2014). Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective. Procedia Technology, 12, 529–534. https://doi.org/10.1016/j.protcy.2013.12.525

[11]. Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29(1), 84–106. https://doi.org/10.1016/j.future.2012.05.023

[12]. Chikin, A., Gobran, T., & Amaral, J. N. (2019). OpenMP Code Offloading: Splitting GPU Kernels, Pipelining Communication and Computation, and Selecting Better Grid Geometries. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11381 LNCS, 51–74. https://doi.org/10.1007/978-3-030-12274-4_3

[13]. Mell, P., & Grance, T. (2011). The NIST-National Institute of Standars and Technology-Definition of Cloud Computing. NIST Special Publication 800-145, 7.

[14]. Mora, H. M., Gil, D., López, J. F. C., & Pont, M. T. S. (2015). Flexible framework for real-time embedded systems based on mobile cloud computing paradigm. Mobile Information Systems, 2015. https://doi.org/10.1155/2015/652462

[15]. Colom, J. F., Gil, D., Mora, H., Volckaert, B., & Jimeno, A. M. (2018). Scheduling framework for distributed intrusion detection systems over heterogeneous network architectures. Journal of Network and Computer Applications, 108(July 2017), 76–86. https://doi.org/10.1016/j.jnca.2018.02.004

[16]. Pravneet Kaur and Gagandeep (2019). Computational Offloading Paradigms in Mobile Cloud Computing Issues and Challenges. In Advances in Big Data and Cloud Computing, Advances in Intelligent Systems and Computing (Vol. 750). https://doi.org/10.1007/978-981-13-1882-5_8

[17]. Pederson, M.V., Fitzek, F.H.P (2012): Mobile clouds: the new content distribution platform. In: Proceeding of IEEE, vol. 100, no. Special Centennial Issue, pp. 1400–1403.

[18]. J. Flinn, S. Park, M. Satyanarayanan, (2002), Balancing performance, energy, and qual- ity in pervasive computing, in: Proceedings ofthe 22nd International Confer- ence on Distributed Computing Systems, 2002, IEEE, 2002, pp. 217–226.

[19]. R. Balan, M. Satyanarayanan, S. Park, T. Okoshi (2003), Tactics-based remote execution for mobile computing, in: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, ACM, 2003, pp. 273–286.

[20]. E. E. Marinelli (2009), Hyrax: cloud computing on mobile devices using MapReduce, Masters Thesis, Carnegie Mellon University, 2009.

[21]. G. Huerta-Canepa, D. Lee (2010), A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing& Services: Social Networks and Beyond, MCS'10, ACM,NewYork, NY, USA, 2010, pp. 6:1–6:5.

[22]. R. Kemp, N. Palmer, T. Kielmann, H. Bal (2010), Cuckoo: a computation offloading framework for smartphones, in: Proceedings of The Second International Conference on Mobile Computing, Applications, and Services, MobiCASE'10.

[23]. L. Deboosere, P. Simoens, J.D. Wachter, B. Vankeirsbilck, F.D. Turck, B. Dhoedt, P. Demeester, Grid design for mobile thin client computing, Future Generation Computer Systems 27 (2011) 681–693.

[24]. Biswas, M., & Whaiduzzaman, M. (2018). Efficient Mobile Cloud Computing through Computation Offloading. International Journal of Advancements in Technology, 10(1), 1–7. https://doi.org/10.4172/0976-4860.1000225.

[25]. Ibrahim, S., Jin, H., Cheng, B., Cao, H., Wu, S., & Qi, L. (2009). CLOUDLET: Towards mapreduce implementation on virtual machines. Proc. 18th ACM International Symposium on High Performance Distributed Computing, HPDC 09, Co-Located with the 2009 International Symposium on High Performance Distributed Computing Conf., HPDC'09, 65–66. https://doi.org/10.1145/1551609.1551624

[26]. Zhou, Bowen. "Code Offloading and Resource Management Algorithms for Heterogeneous Mobile Clouds". Último acceso: 02/01/2020.

[27]. Latifa, E. R., & Ahmed, E. K. M. (2016). Android: Deep look into Dalvik VM. Proceedings of the 2015 5th World Congress on Information and Communication Technologies, WICT 2015, 35–40. https://doi.org/10.1109/WICT.2015.7489641

[28]. Cuervoy, E., Balasubramanian, A., Cho, D. K., Wolman, A., Saroiu, S., Chandra, R., & Bahlx, P. (2010). MAUI: Making smartphones last longer with code offload. MobiSys'10 - Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, 49–62. https://doi.org/10.1145/1814433.1814441

[29]. Gordon, M. S., Anoushe Jamshidi, D., Mahlke, S., Morley Mao, Z., & Chen, X. (2012). CoMET: Code offload by migrating execution transparently. Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2012, 93–106.

[30]. D. Huang, X. Zhang, M. Kang, J. Luo (2010), Mobicloud: building secure cloud framework for mobile computing and communication, in: Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering, SOSE, pp. 27–34.

[31]. G. Schaffrath, S. Schmid and A. Feldmann (2012), "Optimizing Long-Lived CloudNets with Migrations," 2012 IEEE Fifth International Conference on Utility and Cloud Computing, Chicago, IL, 2012, pp. 99-106.

[32]. Biswas, M., & Whaiduzzaman, M. (2018). Efficient Mobile Cloud Computing through Computation Offloading. International Journal of Advancements in Technology, 10(1), 1–7. https://doi.org/10.4172/0976-4860.1000225

[33]. Thompson, B., & Morris-King, J. (2018). An agent-based modeling framework for cybersecurity in mobile tactical networks. Journal of Defense Modeling and Simulation, 15(2), 205–218. https://doi.org/10.1177/1548512917738858

[34]. Goyal, P., Parmar, V., & Rishi, R. (2011). MANET : Vulnerabilities, Challenges, Attacks, Application. 11(January), 32–37.

[35]. Anjum, M. N., Chowdhury, C., & Neogy, S. (2019). Implementing mobile agent based secure load balancing scheme for MANET. 2019 International Conference on Opto-Electronics and Applied Optics, Optronix 2019, 1–6. https://doi.org/10.1109/OPTRONIX.2019.8862375

[36]. Alsboui, T., Qin, Y., Hill, R., & Al-Aqrabi, H. (2020). Enabling distributed intelligence for the Internet of Things with IOTA and mobile agents. Computing. https://doi.org/10.1007/s00607-020-00806-9

[37]. Cao, Z., Zhou, P., Li, R., Huang, S., & Wu, D. (2020). Multi-Agent Deep Reinforcement Learning for Joint Multi-Channel Access and Task Offloading of Mobile Edge Computing in Industry 4.0. IEEE Internet of Things Journal, 4662(c), 1–1. https://doi.org/10.1109/jiot.2020.2968951

[38]. Salkenov, A., & Bagchi, S. (2019). Cloud based autonomous monitoring and administration of heterogeneous distributed systems using mobile agents. Future Generation Computer Systems, 99, 527–557. https://doi.org/10.1016/j.future.2019.04.047

[39]. Sharma, S. K., & Wang, X. (2017). Live Data Analytics with Collaborative Edge and Cloud Processing in Wireless IoT Networks. IEEE Access, 5, 4621–4635. https://doi.org/10.1109/ACCESS.2017.2682640

[40]. Ouahmane, H., Kartit, A., & Marwan, M. (2018). A Secured Data Processing Technique for Effective Utilization of Cloud Computing. Journal of Data Mining & Digital Humanities, Special Is.

[41]. Xia, Z., Ma, X., Shen, Z., Sun, X., Xiong, N. N., & Jeon, B. (2018). Secure image LBP feature extraction in cloud-based smart campus. IEEE Access, 6, 30392–30401. https://doi.org/10.1109/ACCESS.2018.2845456

[42]. Lima, V. B., & Maniyath, S. R. (2018). Geometric location finder based on encrypted spatial data using geometric range queries. Proceedings - 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2018, 75–79. https://doi.org/10.1109/ICDI3C.2018.00024

[43]. Zhang, Y., Xiao, X., Yang, L. X., Xiang, Y., & Zhong, S. (2020). Secure and Efficient Outsourcing of PCA-Based Face Recognition. IEEE Transactions on Information Forensics and Security, 15, 1683–1695. https://doi.org/10.1109/TIFS.2019.2947872

[44]. Kalideen, M. R., & Tugrul, B. (2018). Outsourcing of secure k-nearest neighbours interpolation method. International Journal of Advanced Computer Science and Applications, 9(4), 319–323. https://doi.org/10.14569/IJACSA.2018.090446

[45]. Bao, W., & Li, Q. (2018). Efficient Privacy-Preserving Outsourcing of Large-Scale Geometric Programming. Proceedings - 2018 2nd IEEE Symposium on Privacy-Aware Computing, PAC 2018, 55–63. https://doi.org/10.1109/PAC.2018.00012

[46]. Danevičius, E., Maskeliunas, R., Damaševičius, R., Połap, D., & Wożniak, M. (2018). A soft body physics simulator with computational offloading to the cloud. Information (Switzerland), 9(12), 1–12. https://doi.org/10.3390/info9120318

[47]. Da Silva, D. W. H. A., Oliveira, H., Chow, E., Barillas, B. S., & De Araujo, C. P. (2019). Homomorphic image processing over geometric product spaces and finite P-adic arithmetic. Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom, 2019-Decem, 27–36. https://doi.org/10.1109/CloudCom.2019.00017

[48]. Li, X., Zhu, Y., Wang, J., & Zhang, J. (2019). Efficient and secure multi-dimensional geometric range query over encrypted data in cloud. Journal of Parallel and Distributed Computing, 131, 44–54. https://doi.org/10.1016/j.jpdc.2019.04.015

[49]. Vo, H., Kong, J., Teng, D., Liang, Y., Aji, A., Teodoro, G., & Wang, F. (2019). MaReIA: a cloud MapReduce based high performance whole slide image analysis framework. Distributed and Parallel Databases, 37(2), 251–272. https://doi.org/10.1007/s10619-018-7237-1

[50]. Lu, X., Kumaravelu, G., & Okwudire, C. E. (2019). An evaluation of data size reduction techniques for improving the reliability of cloud-based CNC for a 3D printer. Procedia Manufacturing, 34, 903–910. https://doi.org/10.1016/j.promfg.2019.06.157

[51]. Wang, Chen, & Lin. (2019). A Collaborative and Ubiquitous System for Fabricating Dental Parts Using 3D Printing Technologies. Healthcare, 7(3), 103. https://doi.org/10.3390/healthcare7030103

[52]. Okwudire, C. E., Lu, X., Kumaravelu, G., & Madhyastha, H. (2020). A three-tier redundant architecture for safe and reliable cloud-based CNC over public internet networks. Robotics and Computer-Integrated Manufacturing, 62(May 2019). https://doi.org/10.1016/j.rcim.2019.101880

[53]. Mentone A., Di Luccio D., Landolfi L., Kosta S., Montella R. (2019) CUDA Virtualization and Remoting for GPGPU Based Acceleration Offloading at the Edge. Internet and Distributed Computing Systems. Lecture Notes in Computer Science, vol 11874. Springer, Cham. https://doi.org/10.1007/978-3-030-34914-1_39

[54]. Ayad M., Taher M., Salem A., (2015) Mobile GPU Cloud Computing with real time application, International Conference on Energy Aware Computing Systems & Applications, https://doi.org/10.1109/ICEAC.2015.7352209

[55]. Di Lauro R., Giannone F., Ambrosio L., Montella R., (2012) Virtualizing General Purpose GPUs for High Performance Cloud Computing: An Application to a Fluid Simulator, IEEE 10th International Symposium on Parallel and Distributed Processing with Applications. https://doi.org/10.1109/ISPA.2012.136

[56]. Seritan S., Thompson K., Martínez T.J., (2020) TeraChem Cloud: A High-Performance Computing Service for Scalable Distributed GPU-Accelerated Electronic Structure Calculations, Journal of Chemical Information and Modeling, 60 (4):2126–2137. https://doi.org/10.1021/acs.jcim.9b01152

[57]. Yi, X., Duan, J., & Wu, C. (2017). GPUNFV: A GPU-accelerated NFV system. ACM International Conference Proceeding Series, 85–91. https://doi.org/10.1145/3106989.3106990

[58]. Lallet, J., Enrici, A., & Saffar, A. (2018). FPGA-Based System for the Acceleration of Cloud Microservices. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB, 2018-June. https://doi.org/10.1109/BMSB.2018.8436912

[59]. Ojika, D., Gordon-ross, A., Lam, H., Patel, B., Kaul, G., Strayer, J., Dell, F., & Corporation, E. M. C. (2018). Using FPGAs as Microservices: Technology, Challenges and Case Study. Bpoe-9 @ Asplos 2018, 0–5. http://prof.ict.ac.cn/bpoe-9/BPOE-9 _ The Ninth workshop on Big Data Benchmarks, Performance Optimization, and Emerging Hardware_files/regular1.pdf

[60]. Jiang, S., He, D., Yang, C., Xu, C., Luo, G., Chen, Y., Liu, Y., & Jiang, J. (2018). Accelerating Mobile Applications at the Network Edge with Software-Programmable FPGAs. Proceedings - IEEE INFOCOM, 2018-April, 55–62. https://doi.org/10.1109/INFOCOM.2018.8485850

[61]. Qingqing, L., Yuhong, F., Pena Queralta, J., Gia, T. N., Tenhunen, H., Zou, Z., & Westerlund, T. (2019). Edge Computing for Mobile Robots: Multi-Robot Feature-Based Lidar Odometry with FPGAs. 2019 12th International Conference on Mobile Computing and Ubiquitous Network, ICMU 2019, 4–5. https://doi.org/10.23919/ICMU48249.2019.9006646

[62.] Chung J-M, Park Y-S, Park J-H, Cho H. (2015) Adaptive Cloud Offloading of Augmented Reality Applications on Smart Devices for Minimum Energy Consumption, Transactions on Internet and Information Systems 9(8): 3092-3102. http://dx.doi.org/10.3837/tiis.2015.08.020

[63]. Liu W., Ren J., Huang G., He Y., Yu G., (2019) Data Offloading and Sharing for Latency Minimization in Augmented Reality Based on Mobile-Edge Computing, IEEE 88th Vehicular Technology Conference (VTC-Fall), http://dx.doi.org/10.1109/VTCFall.2018.8690922

[64]. Zhou P., Zhang X., Braud T., Hui P., Kangasharju J., (2018) ARVE: Augmented Reality Applications in Vehicle to Edge Networks, Proceedings of the 2018 Workshop on Mobile Edge CommunicationsAugust. pp. 25–30. https://doi.org/10.1145/3229556.3229564

[65]. Liu L., Li H., Gruteser M., (2019) Edge Assisted Real-time Object Detection for Mobile Augmented Reality, The 25th Annual International Conference on Mobile Computing and Networking, 25: 1–16. https://doi.org/10.1145/3300061.3300116

[66]. Zhang, W., Liao, X., Li, P., Jin, H., & Lin, L. (2017). ShareRender: Bypassing GPU virtualization to enable fine-grained resource sharing for cloud gaming. MM 2017 - Proceedings of the 2017 ACM Multimedia Conference, 324–332. https://doi.org/10.1145/3123266.3123306

[67]. Zhang W., Liao X., Li P., Jin H., Lin L., Zhou B.B., Fine-Grained Scheduling in Cloud Gaming on Heterogeneous CPU-GPU Clusters, IEEE Network, vol. 32, no. 1, pp. 172-178, 2018.

[68]. Lynn, R., Contis, D., Hossain, M., Huang, N., Tucker, T., & Kurfess, T. (2017). Voxel model surface offsetting for computer-aided manufacturing using virtualized high-performance computing. Journal of Manufacturing Systems, 43, 296–304. https://doi.org/10.1016/j.jmsy.2016.12.005

[69]. Xie, J., Wang, X., Yang, Z., & Hao, S. (2019). An Integrated Cloud CAE Simulation System for Industrial Service Applications. IEEE Access, 7, 21429–21445. https://doi.org/10.1109/ACCESS.2019.2895956

[70]. Raei H., Yazdani N., and Shojaee R., (2017) Modeling and performance analysis of cloudlet in Mobile Cloud Computing, Performance Evaluation, 107: 34–53. https://doi.org/10.1016/j.peva.2016.10.005

[71]. Mora H., Mora Gimeno FJ., Signes-Pont MT, Volckaert B (2019) Multilayer architecture model for mobile cloud computing paradigm, Complexity. Article ID 3951495. https://doi.org/10.1155/2019/3951495

[72]. Nguyen Q-H, Dressler F. (2020), A smartphone perspective on computation offloading—A survey, Computer Communications, 159: 133-154. https://doi.org/10.1016/j.comcom.2020.05.001

[73]. Mijumbi R., Asthana A., Koivunen M., Haiyong F., Zhu Q. (2020) Design, implementation, and evaluation of learning algorithms for dynamic real-time network monitoring, International Journal of Network Management. https://doi.org/10.1002/nem.2108

[74]. Benzaid C., Taleb T., (2020) AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions, IEEE Network, 34 (2): 186 - 194. https://doi.org/10.1109/MNET.001.1900252

[75]. Kijak J., Martyna P., Pawlik M., Balis B., Malawski M., (2018) Challenges for Scheduling Scientific Workflows on Cloud Functions, IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA. https://doi.org/10.1109/CLOUD.2018.00065

[76]. A. Mondal, A. R. Kabbinale, S. Shailendra, H. K. Rath and A. Pal, "PPoS: A Novel Sub-flow Scheduler and Socket APIs for Multipath TCP (MPTCP)," 2018 Twenty Fourth National Conference on Communications (NCC), Hyderabad, 2018, pp. 1-6, https://doi.org/10.1109/NCC.2018.8600192

[77]. V. Tran and O. Bonaventure, "Beyond socket options: making the Linux TCP stack truly extensible," 2019 IFIP Networking Conference (IFIP Networking), Warsaw, Poland, 2019, pp. 1-9, https://doi.org/10.23919/IFIPNetworking.2019.8816857