Research Challenges in Query Processing and Data Analytics on the Edge

Blesson Varghese
Queen's University Belfast, UK
b.varghese@qub.ac.uk

Suprio Ray
University of New Brunswick, Fredericton, Canada
sray@unb.ca

Bhavesh N. Gohil
Sardar Vallabhbhai National Institute of Technology, India
bng@coed.svnit.ac.in

Sergio Vega
Virtalus, USA
sergio@virtalus.com

ABSTRACT

The accelerated growth of data has made efficient query processing and data analytics more important than ever. While the Cloud has provided an excellent underpinning solution to store, manage and process data, it is becoming increasingly difficult, as the Cloud necessitates sending all data that is generated by billions of user devices and sensors to distant data centres far away from the data source. This is expected to make query processing and data analytics challenging. This paper examines the challenges in developing pragmatic solutions for processing queries and performing analytics using the emerging Edge computing paradigm. In this paradigm, compute infrastructure is offered at the edge of the network, which is closer to the data source. A simulation study to highlight the advantages of Edge computing over the Cloud is presented.

ACM Reference Format:
Blesson Varghese, Suprio Ray, Bhavesh N. Gohil, and Sergio Vega. 2019. Research Challenges in Query Processing and Data Analytics on the Edge. In *Proceedings of CASCON'19 (CASCON 2019).* Toronto, Ontario, Canada, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The modern information era is characterized by big data - large volumes of data, with increasing variety and velocity, are generated by sensors, user-devices and gadgets. Commercial jets for example generate 10 terabytes of data for every 30 minutes of flight[1]. Consider environmental sensors that are embedded in an industrial facility within a city. Even with 1,000 such sensors generating data at the rate of 1 record every second, 3.6 million data points are generated every hour and 86.4 billion data points every day. If they were to be processed on the Cloud, then this data will need to be sent through to a remote data store hosted in the Cloud.

Now if a multitude of industrial facilities in a large city or industrial zone uses similar sensors, then it would seem practically impossible to send all the data to the Cloud and process it there. Querying and analyzing this data is an integral part of data management for deriving useful information. The process of retrieving information from a data store based on a set of criteria specified by an end-user is what we refer to in this article as 'query processing'.

Typically, the software pipeline of query processing and analytics that makes use of a traditional distributed database system comprises the following five activities:

i. Data ingestion - the absorption of data from multiple disparate sources for further processing.
ii. Data storage and access - the organization of data in external persistent storage for efficient retrieval.
iii. Data consistency and transaction - synchronization of multiple copies of data when concurrent read/write operations occur (transactional workload).
iv. Retrieval and query execution - generation of efficient plans for executing a query for retrieving all data required.
v. Complex data analytics - execution of statistical and machine learning tasks for extracting knowledge.

These activities pose research and software development challenges for efficiently leveraging distributed data stores that are hosted on the Cloud, which are still being addressed by the community.

It is costly in terms of money and bandwidth to transfer large volumes of data originating from user devices, such as smartphones, or from sensors to geographically distant Cloud data centers for processing and storage. This results in high network traffic and software applications relying on the databases in the Cloud to be less responsive at times. Recently, Google Cloud outages due to network congestion in eastern US was reported[2]. Although using Cloud data centers are sometimes disadvantageous, it may be challenging to find large volumes of storage that is available in the Cloud, elsewhere in a cost-effective manner. Nonetheless, there is a lot of recent conversation on whether processing and storage can be brought nearer to the source.

Future Clouds are anticipated to be organized such that storage and compute capabilities would be moved towards the edge of the network, which we refer to as Edge resources [14, 23, 24]. These resources may include dedicated micro data centers or alternatively, existing resources, such as routers and gateways that are augmented with storage and compute capabilities as shown in Figure 1. It is worth noting that these resources can be meagre in storage, memory and CPU processing capacities when compared to the typical resources available on the Cloud [26]. However, the

[1]https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[2]https://status.cloud.google.com/incident/cloud-networking/19009

CASCON 2019, November 2019, Toronto, Ontario, Canada

ACM ISBN xxx-x-xxxx-xxxx-x/xx/xx.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

desired characteristics of Edge resources are expected to resemble those of the Cloud that would facilitate high-availability of compute and storage, incorporate fault-tolerance at both compute and storage levels, and allow for elastic provisioning of resources to meet workload demands.

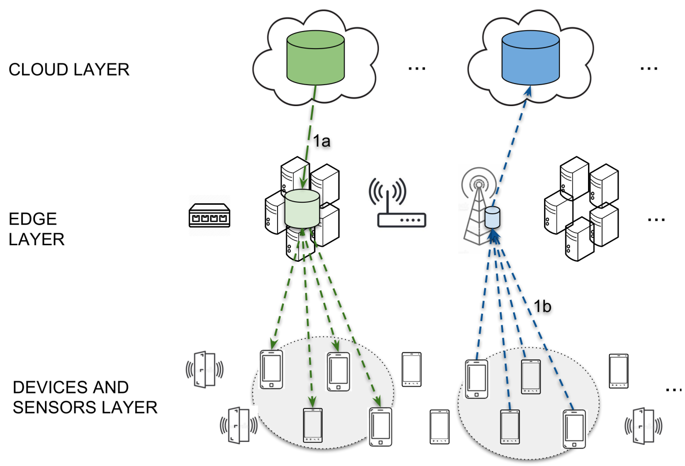


Figure 1: The Edge computing model in which the Cloud and devices and sensors layers interact with Edge resources, such as dedicated micro data centres or Internet nodes, including switches, base stations and routers. The Edge has limited processing and storage capabilities when compared to the Cloud. Software solutions for query processing and data analytics at the Edge will rely on data ingestion, which may take two forms. Firstly, 1a, in which data from the Cloud is partitioned and located at the Edge which is used by end user devices or sensors. Secondly, 1b, in which data is pre-processed at the Edge before sending data to the Cloud store.

It is therefore expected that the challenges involved in achieving the above characteristics on the Cloud will be inherited by the Edge [14, 25]. They may even be aggravated when using the Edge, because they are limited. This paper highlights the challenges in developing practical software solutions for query processing and analytics on the Edge. We also present an experimental study with the EdgeCloudSim [22] simulator to illustrate that the Edge can be used to address the issue of high network latency associated with moving high volumes of data to remote Cloud data centers.

2 CHALLENGES

The approach taken is to discuss the challenges arising from each of the five activities involved in query processing and analytics, namely ingestion, storage and access, consistency and transaction, information retrieval and processing, and complex analytics, as shown in Figure 2. The purpose is to highlight the challenges that the research community will encounter when developing software systems for query processing and analytics on the Edge, rather than providing concrete solutions for tackling these challenges.

2.1 Data Ingestion

The Edge may be utilized by an application in either or both of the following two ways. Firstly, bringing data already on the Cloud towards the Edge to reduce communication latencies for user devices. For example, consider the virtual map used in the PokeMon Go online game. If a user were playing the game in Belfast, then the data related to the map of Belfast may be brought to Edge resources in Belfast, thereby reducing the need for each individual user sending requests and accessing a Cloud data centre in the USA.

Edge resources are inherently heterogeneous (different storage capacities for example) and in comparison to the Cloud are resource constrained (they have limited compute and storage available on them). Therefore, pursuing activities in the software pipeline for query processing and analytics will be challenging.

Partitioning data on the Cloud that will be ingested by the Edge is a key task. Distributed data-stores, such as HBase [2] and Cassandra [1], attempt to partition data evenly among the participating nodes in the Cloud. Such even distribution of data is not suitable for Edge-based data processing, since wide-area network communication may incur significant latency.

A key challenge will be matching the volume of the partitioned data against resources available on the Edge. If sufficient storage space is unavailable on any single Edge resource, then it will need to be further partitioned across different Edge resources. If there is sufficient storage space, then another consideration would be whether all users that will make use of the data can be serviced by a single Edge resource. If that were not possible, then the data will need to be simply replicated across the Edge. However, the trade-offs in partitioning data across the Edge versus replicating data on the Edge remains an open research question.

Secondly, it would be necessary to try to keep data generated by devices and sensors at the Edge closer to the data source, instead of sending all the data to a centralized Cloud facility. High velocity data generated from myriads of mobile devices and sensors are a hallmark of big data and novel approaches may be needed to manage such data [19]. For example, many temperature sensors embedded in an environment may generate temperature readings at a high rate. Given the potential data volume generated by these sensors, it may not be possible to store all temperature readings subsequently in a persistent store. Instead, these data points may be filtered to retain temperature readings that are outliers or periodic measurements. This process can be performed at the Edge, rather than at the central Cloud facility to avoid the movement of high volume and high velocity data over the network [17].

In the above context, two challenges will need to be surmounted. The first is to determine the trade-off between the degree of data abstraction and the availability of storage at the Edge. If a lot of data is filtered out at the Edge, then the learning models of some of the machine learning/data analytics applications or services may not have sufficient data for training. On the other hand, it would not be possible to find storage for large quantities of data at the Edge [21]. The second is to determine the flow control on the Edge for streaming data. It is currently unknown whether the rate of data ingestion needs to be regulated on each individual Edge resource or across all resources for a given application.

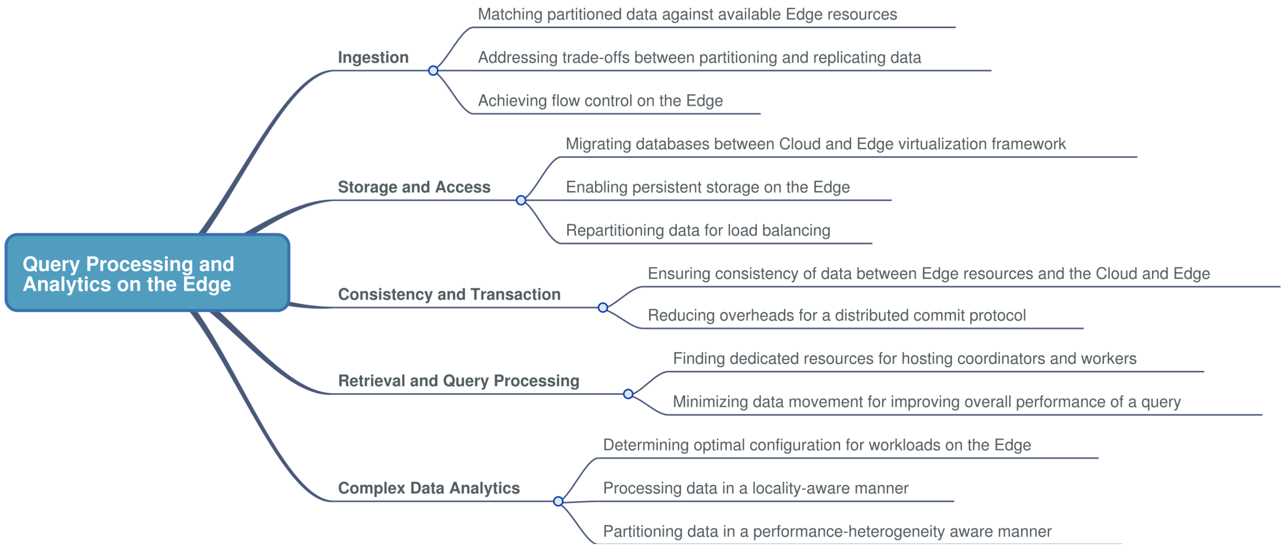


Figure 2: Challenges in developing software systems for query processing and performing data analytics on the Edge

2.2 Data Storage and Access methods

Similar to the Cloud, the Edge is anticipated to provide virtualized resources for supporting multiple users. Virtualization frameworks used in the Cloud and Edge are different. Existing database Cloud services are built on Virtual Machines (VMs) and require migration, replication or cloning [15, 16] of databases to ensure high availability and elastic scalability for applications that rely on them.

Lightweight virtualization, such as container, is preferable at the Edge because individual Edge resources are storage and memory limited when compared to a Cloud server. However, there is no consensus on which container technology might be the most appropriate for using on the Edge.

Containers can be classified into two categories: system containers and application containers. System containers, like LXC/LXD [4] are designed to run a full Linux operating system (OS), similar to running the OS on bare-metal or in a VM. In contrast, application containers, such as Docker [3], focus on ephemeral, stateless and minimum resource consuming containers that typically would not get upgraded or re-configured, but instead replaced entirely. Both approaches have pros and cons when employed at the Edge.

Docker is designed to run a single application. More than one Docker container should be used when running applications comprising a number of micro-services. In this case, different containers should be located on the same physical resource if possible and would require a more complex orchestrator. This is not the case when using LXC/LXD, since they can run several services inside the same container. Also, the data store in Docker containers is located outside and is not a part of the container itself [7]. This becomes a difficult challenge in a dynamic environment like the Edge when a container needs to be migrated from one resource to another, because the container and its data store are physically separate. The migration of containers across heterogeneous Edge resources poses a further challenge for partitioned data at the Edge (highlighted in Figure 3). In a static environment, the data needs to be simply partitioned once. However, in a dynamic Edge environment in which the workloads are changing, the challenge of repartitioning data for efficient load balancing by matching the requirements of the new Edge resource onto which the container migrated will need to be addressed. Hence, a problem that needs to be addressed is how persistent data can be made available.

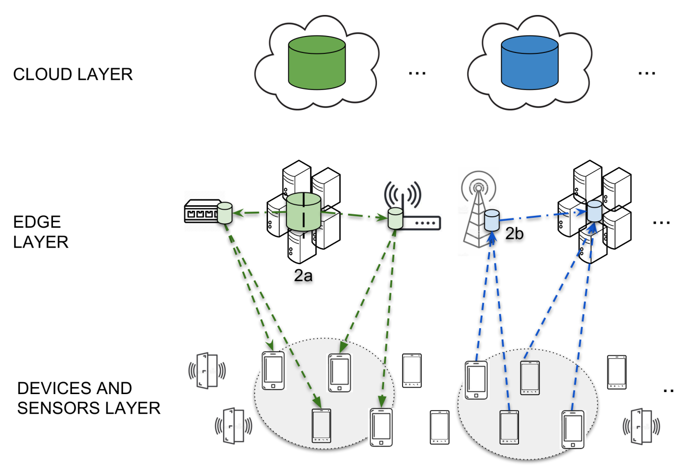


Figure 3: Edge-based software for query processing and data analytics will need to support partitioning, cloning and migration of data across Edge resources. Data residing on Edge resources may need to be: (i) Partitioned as shown in 2a. The partitioned data may be migrated to other Edge nodes, or (ii) Cloned as shown in 2b. The cloned data may reside on another Edge node. The connections with end user devices and sensors are reconfigured. Partitioned data and cloned data need to be consistent and it will be challenging to query multiple data partitions for retrieving data.

LXC/LXD, on the other hand, uses persistent storage inside the container and supports several storage backends like Ceph or ZFS. If the IaaS model is extended to Edge resources, then it may be easier for users to run their applications on a full OS stack. Although LXC/LXD, provides a closer experience to a traditional VM or bare-metal server, Docker is lighter than LXC/LXD in terms of hardware requirements. The Docker layering system enables applications to be deployed faster, reducing bandwidth usage and the storage footprint per container image.

Achieving persistent storage without using significant resources is an open problem. The challenge arising from using different virtualization frameworks at the Cloud and Edge results in the need for novel techniques that enable seamless migration of a virtualized Cloud database service to a container-based Edge service.

2.3 Data Consistency and Transaction

The potentially unlimited storage available on the Cloud allows for hosting large datasets. However, the Edge may only store a subset (related to a specific set of users or to a specific geo-location that provides a local view) of the global data set available in the Cloud. In addition, if the local dataset is large for a single Edge resource, then it may be partitioned across multiple Edge resources. Furthermore, to provide fault tolerance and high availability the local dataset may even be replicated across the Edge. One challenge that will need to be addressed is to ensure that data is consistent between multiple replicas on the Edge. If the Edge data is a subset of a dataset in the Cloud, then consistency will need to be achieved even between the Cloud and the Edge (as shown in Figure 3).

A transaction is a collection of operations, such as reads and writes, on a dataset that need to be performed in an all-or-nothing manner. If even a single operation fails, then none of the operations in the transaction succeeds, which is referred to as *rollback*. If the transaction is successful, then it is *committed*. Each committed transaction must leave the datastore in a consistent state [6]. A transaction may access multiple data partitions stored across different Edge resources. This requires the support for distributed transactions that employ commit protocols to ensure that all Edge resources involved in the transaction either commit or none do. A widely used distributed commit protocol is the two-phase commit (2PC) [13], which requires sending messages among the Edge resources to reach a consensus before committing. Existing protocols employed on Cloud resources suffer from high latency due to the overheads from sending requests over the network and from using virtualized storage [8]. It is still challenging to reduce the overheads. We anticipate that a number of these challenges will be inherited by the Edge. For instance, using current container technology at the Edge, storage is decoupled from the container. This will result in overhead when the container attempts to commit to storage that is located outside the container's Edge resource. This problem will be aggravated from container replication and migration at the Edge to support elasticity. In this context, the 2PC protocol will not be well-suited for a highly dynamic environment involving frequent container migration. The 2PC commit protocol fails to make any progress, if the transaction coordinator encounters failure [10]. Due to the unreliable nature of Edge network and nodes, such failures may be common. Therefore, an adaptive multi-coordinator version of 2PC protocol is necessary. A problem related to consensus protocols, such as 2PC, is that fast event ordering to ensure accuracy in a distributed system. New solutions are needed to opportunistically exploit Edge infrastructure to order events [12].

2.4 Information Retrieval and Query Processing

Data residing on the Edge can be queried for extracting information. There are challenges in query processing when data is replicated and partitioned across multiple Edge resources [11]. We assume structured data stored either in relational databases or data stores supporting schemas. Typically, query processing on partitioned data requires a centralized coordinator that generates an execution plan for each partition. On the Cloud the coordinator can be well placed to manage the query execution of each partition if it were on the Cloud. However, on the Edge it may be harder to find dedicated resources that can host the coordinator for long time periods. It is anticipated that there will be significant competition for acquiring resources on the Edge. The challenge here is to find dedicated Edge resources to host the coordinator as well as allow it to be hosted as a service for prolonged periods of time. A further challenge that arises from placing a centralized query coordinator on the Edge is ensuring that it is fail-safe. In addition, it would be important to address how to deal with the failure of one (or more Edge) resource during the execution of a part of the query plan.

An additional consideration when processing queries at the Edge is the overall performance of the query. One of the factors that affects query performance is data movement between the Edge resources. For instance, if one step in a query plan requires 'materialization' (for example, intermediate results) and this data is required to be moved to the coordinator from an Edge resource, then this will incur significant data transfer overheads. Another factor that affects query performance is resource heterogeneity that leads to the 'straggler effect' [20]. In the context of Edge, this implies that one Edge resource may slow down the entire query processing pipeline. Given that load conditions on the Edge vary over time, the data will need to be repartitioned and distributed to the Edge resources. Otherwise, the straggler effect may become acute. The challenges here are to develop efficient query plans that minimize data movement between Edge resources and to incorporate adaptive query processing techniques for adjusting query plans dynamically by taking the Edge resource load into account.

The Edge computing paradigm assumes the connection of billions of end devices to the Internet. These devices may be mobile and therefore may sporadically appear and disappear within a network. In addition, the devices will operate in a dynamic network topology. Traditional naming schemes used by the coordinator will not be sufficient. However, we note that there are no efficient naming schemes and no standards are available for these [21].

2.5 Complex Data Analytics

It may not be possible to use relational query processing techniques considered above for analyzing large volumes of data that is mostly unstructured or based on loosely defined schemas. Typical use-cases that analyze such data make use of statistical operations and machine learning algorithms. More recently, big data processing frameworks, such as Apache Hadoop [5] and Spark [27] that rely on the MapReduce computing model [9] and distributed file systems have been utilized for this analysis on the Cloud. The MapReduce model comprises map and reduce phases. In the map phase, data is decomposed for processing in parallel, whereas in the reduce phase, intermediate results obtained from the map phase are aggregated. It is already challenging to determine the optimal configuration (for example, the number of compute resources that need to be allocated for the Map and Reduce phases) for a workload in a traditional computing system. This becomes more challenging in the face of changing workloads, elastic provisioning of resources and the competition for limited resources on the Edge.

Since data movement between different phases of the MapReduce model may tend to be expensive on the Edge an additional challenge arises. The challenge is processing data in a locality-aware manner such that a Map operation, for example, is assigned to the same compute resource, which hosts the data partition required by that operation in the distributed file system.

Normally data partitioning approaches, such as range or hash partitioning assign workload, such that each computing resource is allocated roughly an equal volume of data. These approaches can be applied to the Edge, but Edge resources have heterogeneous storage and processing capabilities. Therefore, the challenge that arises is processing data in a performance-heterogeneity aware manner [20]. This can be addressed by recognizing more capable Edge resources and allocating them more work than weaker resources. This in turn may reduce the 'straggler effect'.

Machine learning-based analytic systems have gained popularity on the Cloud, such as recommendation systems for online shopping or financial tools for fraud detection. In these systems, user-specific data is aggregated in the Cloud. One of the major concerns is data privacy and therefore there is suggestion to perform data analytics at the edge. However, to make this possible 'composable' services are required, to support complex data analytic systems at the edge by building on basic machine learning services [28]. However, the key challenges that need to be addressed in this arena include, service composition - constructing a workflow of multiple services using concise domain specific languages, and deployment - mechanisms to deploy services in multiple edge locations or along the Cloud-Edge continuum as required.

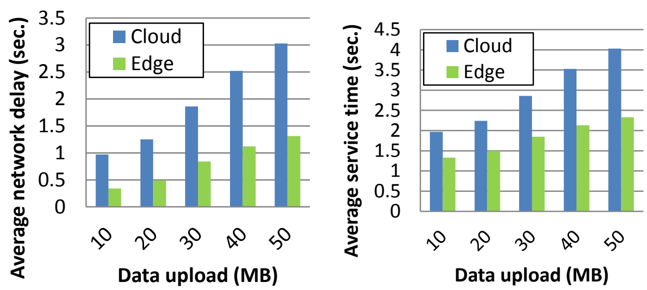
An important category of data analytics applications involve processing large volumes of latency-critical streaming data. These applications include IoT sensor monitoring, video surveillance, augmented reality, and object recognition in driver-less cars. Supporting the real-time requirements of such applications using traditional Cloud-based data stores is impractical due to the end-to-end latency entailed by the wide-area network traversal. Although, Edge computing is promising for such purposes, the challenging operating conditions and dynamic nature of Edge devices call for novel approaches to data management. Since, there is a clear trade-off between supporting latency-criticality and accuracy [18], it would be necessary to autonomously monitor run-time conditions to maintain the desired quality of service.

3 AN EXPERIMENTAL STUDY

In this section, a simulation-based experimental evaluation is presented to demonstrate the potential benefit in addressing some of the challenges associated with query processing and data analytics

Parameters	**Query processing**	**Machine learning**	**Augmented reality**
Data upload	10MB to 50MB	15MB	1.50MB
Task length (MIPS)	100K	1M	2K
WLAN bandwidth	1 Gbps		
WAN bandwidth	1 Gbps		
Hosts in Cloud layer	14		
Hosts in Edge layer	14		
Mobile devices	100 to 500		

Table 1: Simulation parameters for Cloud and Edge layers



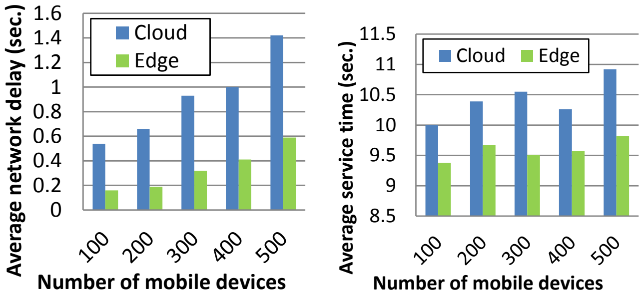
(a) Network delay **(b) Service time**

Figure 4: Network delay and service time of a sample query processing application at the Cloud and Edge

on the Cloud. Although the Edge entails its own set of challenges that are outlined earlier in this paper, this study highlights that the Edge addresses a key issue of the Cloud - network delay (latency), and hence the service time, arising from the movement of large volumes of data.

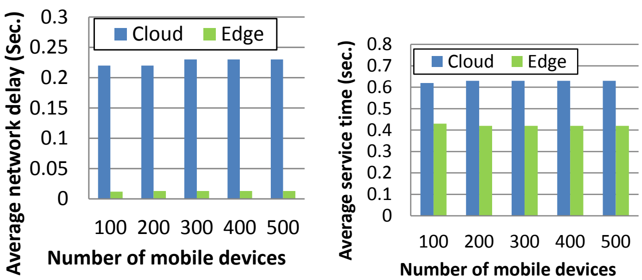
The experiments are carried out on an Edge simulator named EdgeCloudSim [22]. This simulator has a modular architecture and supports network modeling specific to WLAN and WAN, device mobility modeling, and a realistic and tunable load generator. EdgeCloudSim simulates multi-tier scenarios, where multiple Edge servers run in coordination with upper layer Cloud solutions. Three different architecture can be simulated in EdgeCloudSim: (i) single-tier, (ii) two-tier, and (iii) two-tier with an Edge orchestrator (EO). The single-tier architecture allows the mobile devices to utilize the Edge server located in the same building. In the two tier architecture, the mobile devices can offload tasks to the global Cloud by using the WAN connection provided by the connected access point. The two-tier with EO architecture has a considerable advantage, because for the tasks which are executed on the first tier, only the two-tier with EO architecture can offload the tasks to any Edge server located in different buildings. It is assumed that the Edge servers and the EO are connected to the same network.

Three sample applications were simulated with EdgeCloudSim to compare the network delay (latency) and service time when using the Edge and the Cloud. These applications are: query processing, machine learning, and augmented reality, of which the augmented reality application provided by the EdgeCloudSim distribution. These applications are defined by the simulation parameters as shown in Table 1. Figure 4a shows the average network delay of a query processing application on the Edge and Cloud, while



(a) Network delay **(b) Service time**

Figure 5: Network delay and service time of a sample machine learning application at the Cloud and Edge



(a) Network delay **(b) Service time**

Figure 6: Network delay and service time of a sample augmented reality application at the Cloud and Edge

the data upload size is varied from 10MB to 50MB. It is observed that the network delay with the Edge is significantly lower than the Cloud. Also, as shown in Figure 4b, the service time with Edge is considerably lower than that with the Cloud. In Figure 5a and Figure 6a, the average network delay of the machine learning and the augmented reality applications are presented, respectively. In both cases, the number of mobile devices that are connected are varied from 100 to 500. Similar to the previous results, the network delay is significantly lower when using the Edge compared to the Cloud. Moreover, the service times of the machine learning and the augmented reality applications (shown in Figure 5b and Figure 6b respectively) are lower when using the Edge than the Cloud. These preliminary results provide indications that the Edge can be a compelling computing model for data analytics application and query processing. In these experiments, the connections from mobile devices to the Cloud were configured to be reliable. However, in real life scenarios this may not be the case, especially when there is the need to transmit a large volume of data.

4 CONCLUSION

Query processing and data analytics are integral tasks for software systems that make sense of big data. Bringing these tasks to the edge of the network can reduce the volume of data that is sent to a Cloud data center. Using a simulation study we highlight that the Edge can be attractive for query processing and performing analytics, due to significantly lower network delay than the Cloud. At the same time, the challenges in executing software systems on the Edge can be aggravated by an inherently heterogeneous, resource constrained and dynamic Edge when compared to the Cloud. Additionally, there are novel challenges introduced by the Edge that need to be addressed to support highly efficient query processing and data analytics.

REFERENCES

[1] [n. d.]. Apache Cassandra. http://cassandra.apache.org/.
[2] [n. d.]. Apache HBase. https://hbase.apache.org/.
[3] [n. d.]. Docker Containers. https://www.docker.com/.
[4] [n. d.]. Linux Containers. https://linuxcontainers.org/.
[5] Apache Hadoop [n. d.]. Apache Hadoop. http://hadoop.apache.org/.
[6] Philip A Bernstein, Vassos Hadzilacos, and Nathan Goodman. 1987. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley.
[7] Containers [n. d.]. Why Containers Miss a Major Mark: Solving Persistent Data in Docker. https://storageos.com/why-containers-miss-a-major-mark-solving-persistent-data-in-docker/.
[8] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield. 2008. Remus: High Availability via Asynchronous Virtual Machine Replication. In *USENIX Symposium on Networked Systems Design and Implementation.* 161–174.
[9] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM* 51, 1 (Jan. 2008), 107–113.
[10] Jim Gray and Leslie Lamport. 2006. Consensus on Transaction Commit. *ACM Transactions on Database Systems* 31, 1 (March 2006), 133–160.
[11] Harshit Gupta, Zhuangdi Xu, and Umakishore Ramachandran. 2018. DataFog: Towards a Holistic Data Management Platform for the IoT Age at the Network Edge. In *USENIX Workshop on Hot Topics in Edge Computing.*
[12] Zijiang Hao, Shanhe Yi, and Qun Li. 2018. EdgeCons: Achieving Efficient Consensus in Edge Computing Networks. In *USENIX Workshop on Hot Topics in Edge Computing.*
[13] Rachael Harding, Dana Van Aken, Andrew Pavlo, and Michael Stonebraker. 2017. An Evaluation of Distributed Concurrency Control. *The VLDB Journal* 10, 5 (2017), 553–564.
[14] Cheol-Ho Hong and Blesson Varghese. 2019. Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. *Comput. Surveys* (2019).
[15] Umar Farooq Minhas, Shriram Rajagopalan, Brendan Cully, Ashraf Aboulnaga, Kenneth Salem, and Andrew Warfield. 2013. RemusDB: Transparent High Availability for Database Systems. *The VLDB Journal* 22, 1 (Feb. 2013), 29–45.
[16] Michael J. Mior and Eyal de Lara. 2011. FlurryDB: A Dynamically Scalable Relational Database with Virtual Machine Cloning. In *ACM International Systems and Storage Conference.* 1:1–1:9.
[17] Ioannis Psaras, Onur Ascigil, Sergi Rene, George Pavlou, Alex Afanasyev, and Lixia Zhang. 2018. Mobile Data Repositories at the Edge. In *USENIX Workshop on Hot Topics in Edge Computing.*
[18] Arun Ravindran and Anjus George. 2018. An Edge Datastore Architecture For Latency-Critical Distributed Machine Vision Applications. In *USENIX Workshop on Hot Topics in Edge Computing.*
[19] Suprio Ray, Rolando Blanco, and Anil K. Goel. 2017. High Performance Location-based Services in a Main-memory Database. *GeoInformatica* 21, 2 (Apr 2017).
[20] Suprio Ray, Bogdan Simion, Angela Demke Brown, and Ryan Johnson. 2013. A Parallel Spatial Data Analysis Infrastructure for the Cloud. In *ACM International Conference on Advances in Geographic Information Systems.* 284–293.
[21] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal* 3, 5 (2016), 637–646.
[22] C. Sonmez, A. Ozgovde, and C. Ersoy. 2017. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. In *IEEE International Conference on Fog and Mobile Edge Computing.* 39–44.
[23] Blesson Varghese and Rajkumar Buyya. 2018. Next Generation Cloud Computing: New Trends and Research Directions. *Future Generation Computer Systems* 79 (2018), 849 – 861.
[24] Blesson Varghese, Philipp Leitner, Suprio Ray, Kyle Chard, Adam Barker, Yehia Elkhatib, Herry Herry, Cheol-Ho Hong, Jeremy Singer, Posco Tso, Eiko Yoneki, and Mohamed-Faten Zhani. 2019. Cloud Futurology. *IEEE Computer* 52, 9 (2019).
[25] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S Nikolopoulos. 2016. Challenges and Opportunities in Edge Computing. In *IEEE International Conference on Smart Cloud.* 20–26.
[26] Nan Wang, Blesson Varghese, Michalis Matthaiou, and Dimitrios S. Nikolopoulos. 2018. ENORM: A Framework For Edge NOde Resource Management. *IEEE Transactions on Services Computing* (2018).
[27] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *USENIX Workshop on Hot Topics in Cloud Computing.*
[28] Jianxin Zhao, Tudor Tiplea, Richard Mortier, Jon Crowcroft, and Liang Wang. 2018. Data Analytics Service Composition and Deployment on Edge Devices. In *Workshop on Big Data Anal. and Machine Learning for Data Com. Networks.*