Ad-hoc Edge Cloud: A framework for dynamic creation of Edge computing infrastructures

Ana Juan Ferrer
NG Cloud Lab Reserch & Innovation , Computer Sciences, Multimedia and Telecomunication Studies
Atos, Universitat Oberta de Catalunya
Barcelona, Spain
ana.juanf@atos.net, ajuanf@uoc.edu

Joan Manuel Marqués, Josep Jorba
Computer Sciences, Multimedia and Telecomunication Studies
Universitat Oberta de Catalunya
Barcelona, Spain
jmarquesp@uoc.edu, jjorbae@uoc.edu

Abstract— **Over the course of the last decades there has been significant growth in smartphone penetration and capacities. This trend presently complements the rise of IoT and ever complex and smarter connected devices. On balance, considerable contribution has been made to the emergence of unprecedented computing capacity at the edge of the network, which is only expected to have a long-lasting impact with the rise of connected vehicles, robots and drones. Considering the abovementioned context, computing will cease to be confined to certain devices located into large data centers or stationary edge devices, instead it will be embedded and pervasive to virtually everything. This paper provides a comprehensive framework for distributed and decentralized Edge Computing systems dynamically formed out of edge computing resources with the objective of harnessing the increasingly available computing at the edge. In order to facilitate this, we propose an architecture for this Ad-hoc Edge Computing infrastructure. Furthermore, two critical aspects for Resource Management are evaluated in the present framework: namely scalability and implications of node volatility.**

Keywords— Ad-hoc Edge Cloud, Ad-hoc Edge Architecture, Edge Computing, Ad-hoc Edge Computing infrastructure

I. Introduction and Motivation

Computing plays a crucial part in our everyday reality. The development of Moore's law throughout the last decades has allowed us to have computation at the palm of our hands on our smartphones. Owing to their help have brought us to computing and provided us with new ways to utilize applications and services.

The appearance of IoT is also expected to boost the number of connected devices worldwide, from the billions of units existing today, to the tens of billions of units highly likely to be deployed in the coming years. The development of IoT increases the need of processing close to the data sources. The principal aim of Edge Computing (as named by major Cloud providers AWS[1] and Azure [2]) is to tackle this issue by providing a novel paradigm which extends capacities of Cloud Computing to the edge of the network avoiding present latency constraints. Edge Computing combines networking with typical cloud principles with the intention to create distributed computing platforms to address the specific needs of IoT [3].

Moreover, the materialization of IoT is conducive to the presence of computing not only on our smartphones, but also on a wide diversity of devices (cars, televisions, cameras, etc.). The aforementioned devices are rapidly gaining complexity and converting additional computing capacities into de facto edge devices. IDC[4] predicts that "By 2022, over 40% of organizations' cloud deployments will include edge computing and 25% of endpoint devices and systems will execute AI algorithms.". Some examples of extant market developments which pave the path towards supporting this trend are provided: NVIDIA's Jetson systems for processing on-board edge devices [5]; and Intel Aero Compute Board for UAVs [6]. Furthermore, the rising data availability resulting from IoT deployments together with the recent advances and popularity of AI and Deep learning at the Edge is expected to increase computing demand by several orders of magnitude. Thereupon, Smartphones are recognized as the forerunners of the Edge device. Undoubtedly, these devices will be enhanced with additional Edge devices such as smart speakers, automotive, drones, consumer and enterprise robots or security cameras [7]. Thus, computing will cease to be constricted to specific devices, instead it will be virtually embedded and pervasive to everything[8].

Current Edge computing developments regard the Edge as stationary single device environments which provide computing and storage services to a set of IoT devices located in the vicinity. In these, IoT devices are solely viewed as sources of data, and their increasing complexity in terms of computing and storage capacity is disregarded. However, these devices are progressively drawing on noteworthy resources, and it points to an unjustifiable waste to employ them as data sources of resource richer computing environments. This is additionally exacerbated by the slow progress expected for Moore's Law in the future, which further demands benefitting from all computing capacity available everywhere[9].

This paper formulates a novel a novel Edge computing infrastructure management system, the Ad-hoc Edge Cloud, formed by available edge resources with the objective of

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

harnessing the increasingly accessible computing capabilities from complex IoT devices at the Edge of the network. The overall objective is to respond to the rising demands for processing by means of exploiting the existing capacity in order to extract the value out of the massive amount of data collected at the Edge. The remainder of the paper is organized as follows. Section II elaborates on the envisaged characteristics for Ad-hoc Edge Computing. Section III illustrates the proposed architecture by providing insights about modules and their interactions. Section IV details performed evaluation. Finally, section V presents related work while section VI supplies information regarding future work and conclusions.

II. AD-HOC EDGE COMPUTING CHARACTERISTICS

The unprecedented growth we are witnessing nowadays in the number connected devices calls for harnessing the idle capacity at the Edge of the network. Connected devices are not only permeating everywhere, but also significantly gaining complexity (such as robots and autonomous vehicles). These edge devices can be progressively regarded as Edge devices which provide complex aggregations of computing and storage resources together with diverse and heterogeneous sensors and actuators [8]. In the given context we present Ad-hoc Edge Cloud, a framework in which increasing compute capacity at the Edge can be exploited in a distributed manner by enabling ad-hoc formation of Edge infrastructures created out of available edge devices. Edge resources pose specific challenges for their management as a result of their expected massive number, resource constrains and the fact that in the majority of the cases they are mobile.

The target edge devices for this Ad-hoc Edge Cloud constitute limited resources in terms of computational power and storage capacity. Producers of these edge devices, such as automobiles, drones or robots, are placed under pressure to provide more intelligent capabilities at commercially viable costs. The mobile nature of these devices makes them rely on restricted capacity batteries for energy supply. Enhancing edge devices' on-board computational and storage resources, increases their cost and energy demand, therefore resulting in reduced device autonomy. Hence, any framework aiming to operate in such environment needs to contemplate these limitations. Thereby, Ad-hoc Edge infrastructure management components will require very lightweight implementations which do not hinder the normal functioning of the target edge devices.

In addition, the mobility of the edge devices which form this computing infrastructure leads to a significant breakdown of existing resource management practices in Cloud and data center, as far as connectivity instabilities are concerned. Furthermore, they are affected by specific factors, namely the need for energy optimization and battery lifetime, which can exert negative influence on the availability of these resources.

Another significant factor which differentiates resource management in Ad-hoc Edge Computing from traditional Cloud resource management practices, is the high degree of heterogeneity of the edge devices that can be involved in the infrastructure. In Cloud computing large farms of homogeneous resources provided by a single operator allow for automated management and economies of scale. In a different way our

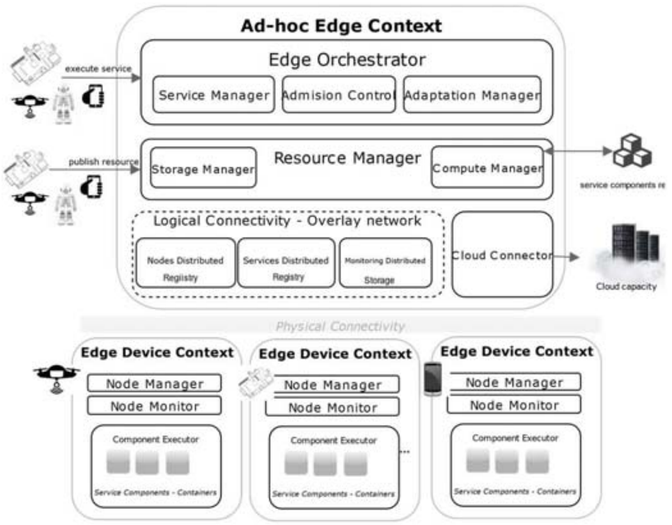

Fig. 1 Ad-hoc Edge Computing Architecture

work needs handle the diversity of edge devices that take part in the infrastructure. Diversity undoubtedly stems from the variety of capacities of the edge resources intended to be used, as well as from aspects such as: supported operating systems (i.e. Linux, Raspbian, Robot Operating system), connectivity protocols (i.e WLAN, Ethernet) and processing architectures (i.e. CPU and GPU).

In view of the above, edge resource management Ad-hoc Edge Computing infrastructure reflects the need to support the operation in heterogeneous constrained devices and dynamic behavior in relation to resource availability.

The dynamicity present in resource availability is commonly referred to as node churn. Node churn describes the dynamic behavior of resources appearing and disappearing from the system. Node churn stresses the need for the Ad-hoc Edge Computing infrastructure to manage uncertainty on resource availability. Edge resources can suddenly appear or disappear from the environment triggered by a change of edge device locations, they can deplete the battery or be affected by many factors which influence the availability as part of the infrastructure. This imposes specific requirements on admission control and service management procedures of the Ad-hoc Edge Computing infrastructure.

Edge resource node churn and resource volatility also heighten the need to provide a decentralized management system to the Ad-hoc Edge Cloud, in order to avoid a single point of failure. It equally aims at ensuring its autonomy by not relying on external management layers located i.e. at the Cloud level, potentially hampering the infrastructure operation in case of losing external connectivity to the cloud. Thereby, the Ad-hoc Edge Computing infrastructure management processes need to be managed in a distributed manner among all available Edge resources.

III. AD-HOC EDGE COMPUTING ARCHITECTURE

Fig. 1 presents the suggested architecture for Ad-hoc Edge Infrastructure. The architecture is structured into two main contexts or building blocks. It is noteworthy recalling that

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

contexts do not represent layers. This is performed with the objective of evidencing that the architecture is not built in a stack-like manner, but that modules and contexts can be executed in any of the Edge Device nodes. Contexts define the separation of concerns with regard to the management of the particular Edge node, Edge Device Context; and Ad-hoc Edge Cloud Context, denoting the overall infrastructure cluster formed out of all participant Edge devices.

A. Components

1) Edge Device Context

The Edge Device context modules enable the Edge resources to execute services or parts of them (services' components).

Node Manager: The Node manager component oversees the interface in order to manage a node, handling the resources at the Edge device. The main aim of this component is to deliver a unified description for the Edge resource characteristics and capabilities, and their offered capacity to the rest of the infrastructure. The Node manager develops an abstraction layer for all types of resources in the infrastructure which permits heterogeneous resources to be handled uniformly in the Ad-hoc Edge infrastructure.

Component Executor: This component provides the means to perform actions related to the life-cycle of application components. The workload virtualization is based on containers, which facilitate the unified execution in heterogeneous execution environments in a variety of Edge devices. The most popular containerization system, Docker [10] is now present in diverse constrained environments such as Raspbian Raspberry Pi Operating System [11], or Robot Operating System [12] as well as specific network devices [13]. Docker is increasingly being complemented with even more lightweight implementations of the containerization technologies among which are included Unikernels [14], Kata Container [15] and gVisor [16]. This permits us to predict the feasibility of our approach in even more constrained execution environments to those able to support Docker today.

Node Monitor: It collects monitoring information about the status of the Edge node. The compiled parameters include the following aspects: physical infrastructure (memory, CPU usage and available storage) bandwidth (connection type and transmission rate), as well as, battery level status. While bandwidth parameters provide a clear understanding of the quality offered by the node to the rest of the Ad-hoc infrastructure, the battery level is particularly important as a factor indicating probability of churn, therefore with the potential of affecting the availability of the node in the infrastructure.

2) Ad-hoc Edge Context

Modules in the Ad-hoc Edge Context offer the functionalities which enable the overall infrastructure management. They warrant the handling of all participant Edge resources as a cluster. A crucial module in this context is the **Logical Connectivity Layer.** It creates and maintains a distributed registry among all participant Edge nodes, allowing for building distributed indexes. The Logical Connectivity layer offers the mechanisms to handle two of the main challenges in the Ad-hoc Edge Infrastructure: distributed management over all available nodes, in order to manage scale; and resource volatility, owing to the probability of node churn. Three distributed indexes are illustrated in this work: **Nodes Distributed Registry**, which supports the storage of the information about the physical resources of Edge nodes added to the system; **Services Distributed Registry**, granting access to information of services in the system; and **Monitoring and Accounting Distributed Storage**, which collects information of node and service execution status and resources consumption.

The enabling mechanism for these modules are distributed key stores. These store the correspondence between a key and a value, similarly to traditional hash tables, running in a distributed system in which look up and storage are scattered among nodes with no central management. Instead, each node maintains a portion of the information along with pointers to the ranges of keys available in other nodes of the distributed storage. Any read, write or look-up operation has to localize, by dint of distributed storage mechanisms, the node in charge of this portion of the key. In Ad-hoc Edge Cloud infrastructure, each Edge node is responsible for a certain part of the overall information system. Each node stores information about resources, services components in execution and monitoring information, therefore without a centralized management as a single point of failure. Distributed key stores offer data distribution and replica mechanisms that supports information recovery in the case of a node abandoning the system, and permit the Ad-hoc Edge Infrastructure to manage node churn at information level.

Resource Manager: Resource manager component allows resources to be published into the Ad-hoc Edge Infrastructure. It provides the resource specification in terms of device characteristics and capacity. The registration of the node is obtained by means of its incorporation to the Nodes Distributed Registry. The publication of a resource incorporates it to the set of resources to be used in the Ad-hoc Edge Infrastructure. This requires the new node to be bootstrapped into the distributed storage by generating a unique id, key, which identifies the Node in the Registry.

The Resource Manager also plays an essential role at service deployment time. Resource Manager offers the interface to the Edge Orchestrator to locate nodes selected as part of Admission control process. Within the Resource Manager, the **Compute Manager** component is responsible for controlling Compute resources in the Edge infrastructure; while **Storage Manager** handles Block storage resources. The evaluation of distributed storages as the enabling mechanisms for Resource Manager and Nodes Distributed Registry is provided in Section IV.

Admission Control: Admission control supplies the necessary mechanisms which allow the decision-making regarding the acceptance or rejection of a service to be executed into the Ad-hoc Edge infrastructure. The challenge of the Admission control component is to select the set of resources that offer sufficient capacity to execute the service, but also to favor those which offer a more stable execution environment in order to handle the environment dynamicity due to node churn. The Admission Control receives the requirements of the service to be executed from the Edge Orchestration in terms of CPU, memory and storage of its components. The Admission Control

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

obtains up- to -date information about available nodes in the Edge infrastructure. Admission Control later performs a filtering and prioritization process among the available Edge nodes in order to select the candidates able to host a service. Filters represent the set of parameters connected with the capacity of the node. These help to determine whether the Edge device host is able or not to host the workload for a minimum period of time. The filter parameters cover: Capacity, as function of (Memory, CPU and Storage) and available percentage of battery. Once the initial filtering process is performed, the remaining nodes are prioritized according to Ranker parameters. Rankers are viewed as the parameters which measure the quality of the node and its stability. They aim to determine or estimate the QoS provided by the edge device host. Among them, we intend to select Edge devices endowed with longer connection times and better battery levels with the aim of minimizing node churn. The result of the admission control process is the assignment of each Service Component to one or a set of Edge Nodes.

Service Manager: This module empowers the management of service execution in remote nodes via the Node Manager interfaces interacting with Services Distributed Registry and Monitoring Distributed Storage. Thus, it obtains information regarding the status of nodes and services controlling their availability and performance. The Service Manager is in charge of performing the operational actions (start, stop, resume and terminate) for the complete service. The Service Manager locates, via the Resource Manager, the node(s) responsible for the execution of a certain service component. Once located, it interacts with the correspondent Node Manager to implement the required operational action on each node. In case of remote node failure (i.e. due to node churn), it handles, together with the Adaptation Engine, the service re-collocation in another available node. In addition to this, the continuous monitoring process can entail other adaptation actions as a result of the need to scale-up or down the number of instances of a component of a service.

Adaptation Engine: This module works in close cooperation with the Service Manager which performs the necessary adaptation actions suggested by it. In the event of adding a new Service Component instance to the execution of a service, it relies on the Admission Control which performs the placement decision for this new component.

Edge Orchestrator: This component constitutes the entry point to execute a service in the Ad-hoc Edge infrastructure. It receives the Service template which provides a deployment specification of a service to be executed in the Ad-hoc Edge infrastructure. The Edge Orchestrator coordinates the deployment of the different Service components interacting with Admission Control to obtain placement alternatives. Subsequently, via the Resource Manager, it implements the service deployment and creates the Service Manager for the Service.

IV. EVALUATION

Validation of the Ad-hoc Edge Computing Infrastructure work has centered on the analysis of the behavior of the **Resource Management** component in situations of massive and dynamic incorporation and removal of Edge nodes. The driving ambition has been to understand how dynamicity on resource availability must be handled in the Ad-hoc Edge Infrastructure. In order to do so, we have relied on Etcd [17], a well-known distributed key storage employed in many distributed systems including Kubernetes [18]. Specifically, we have analyzed its behavior and resource consumption in highly dynamic situations when installed over constrained Edge devices represented by Raspberry Pis. In order to extrapolate our results to a scale not achievable in our existing physical Edge cluster, we have developed a simulation environment in Amazon Web Services [19]. This has also served us to compare the outcomes with another distributed storage system, Apache Cassandra [20] taking into account its Distributed Hash Table architecture.

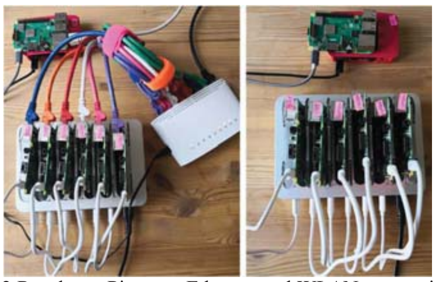
Fig. 2 Raspberry Pi set-up Ethernet and WLAN connectivity

A. Lab Evaluation

Our available physical testbed is depicted in Fig. 2 and it is composed of hardware of the following characteristics:

- 3 x Raspberry Pi 3 Model B (1.2GHz 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1 and 10/100Mbit/s Ethernet Port) equipped with 8GB 32 GB and 4GB micro-sd memory cards.
- 5 x Raspberry Pi 3 Model B + (1.4 GHz Quad Core ARM Cortex-A53, ARMv8-A (64/32-bit), On Board WiFi 802.11ac Dual Band 2.4GHz & 5GHz, 10/100/1000 Mbit/s Ethernet Port) all equipped with 128GB micro-sd memory cards.

The initial step has consisted of flashing all Raspberry Pis micro-sd memory cards with Raspbian Stretch Lite Kernel version 4.1.4 [21]. Afterwards, in each Raspberry, Docker was installed. In the interest of respecting the constrained nature of Edge devices when executing Nodes Distributed Registry and Resource Manager functionalities Docker containers in this the Edge environment have been limited to use a maximum of 32MB. This parametrization is an extremely constrained execution environment for Ad-hoc Edge infrastructure management processes. Our aim when setting up this outstanding resource limited environment has been to validate our approach in an environment which is lightweight in resources occupied by our runtime, leaving capacity for the execution of external workloads. However, we are fully cognizant of the fact that such constrained runtime parametrization is likely to affect the resultant response times of our results.

The focal points of the evaluation of this environment have been the ability of Nodes Distributed Registry layer to support scale in a timely manner, as well as, analysis of the behavior and response time in dynamic resource volatility scenarios, considering diverse rates of nodes churn for the Resource

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

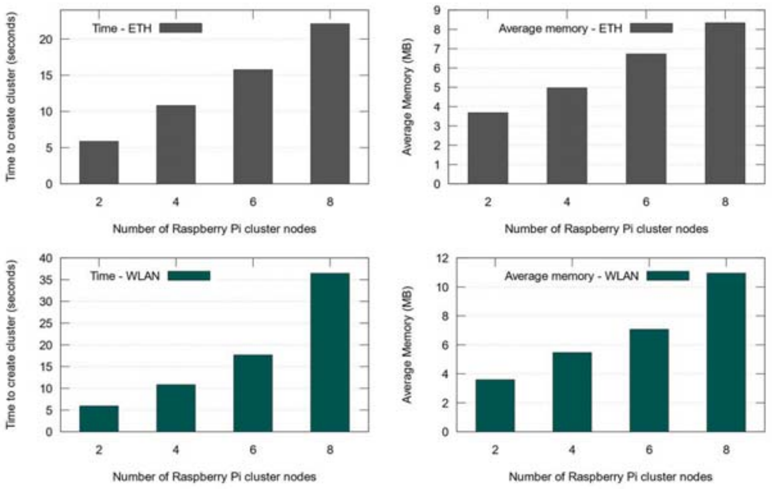

Fig. 3 Scalability Experimentation over Raspberry Pi testbed

Manager. For both of the considered aspects we have drawn a comparison between the results obtained by utilizing the Ethernet connectivity versus the utilization of a Wireless LAN (WLAN) for the same operations. This has helped us to establish a baseline in order to assess the impact of using WLAN connectivity.

Scalability Experimentation

Experimentation concerning scalability aspects has targeted the evaluation of the behavior of Nodes Distributed Registry and Resource Manager to support the relevant addition of Edge nodes and the response times obtained. The process of adding a new node consists of instantiating the Docker container in the Raspberry Pi out of the customized Docker image which contains the distributed storage and Node Resource management code. The new node is registering itself both in the Ad-hoc Edge Computing Infrastructure and Nodes Distributed Registry. By doing so, added resources on the infrastructure become an intrinsic part of it, by means of storing part of the information dedicated to its management.

Scalability experimentation in the physical Raspberry Pi environment has involved all available Raspberry Pis supported by the subsequent creation of clusters of 2, 4, 6 and 8 nodes. In order to do so, the measured aspects have been the necessary time to have an operative Ad-hoc Edge cluster distributed over the selected Raspberry nodes and the resources (memory) these processes consume from the physical nodes. In the execution of these tests we have observed a high degree of variability in response times, due to changing network conditions, therefore this experimentation presents average results obtained by 10 executions of the experiment.

Fig. 3 exhibits tests performed using both Ethernet and WLAN connectivity. In these tests we can detect that on average the use of WLAN connectivity increases the time to create a cluster. The percentage of increment grows in relation with the size of the cluster. For a two-node cluster the difference of creation time is of 1% reaching 64% for an 8 Nodes cluster. The explanation found for this fact is the additional need of data synchronization processes among distributed nodes given that both Ethernet and WLAN experimentation have used the same devices in the same order (with diverse hardware configurations, see A). This observation highlights the need of keeping clusters to the minimal possible granularity at level of constituent number of nodes to consider in a single cluster, or at least the need of taking into account the performance overhead that consideration of larger clusters produces. It is significant to note that differences detected in memory usage of the nodes in the cluster present on average less than 10% variability.

Availability / Churn rates Experimentation

The experimentation on node availability explores the behavior of the Ad-hoc Edge Cloud under diverse churn rates. Namely, once the cluster is set up and a certain number of nodes become suddenly unavailable to the system. The overall intention of this experimentation is to comprehend how a highly dynamic environment with regard to nodes abandoning the system affects the overall performance of Ad-hoc Edge Computing Infrastructure.

In this set of experiments, we measure the time required for the Ad-hoc Edge Computing Infrastructure Nodes Distributed Registry to get to a consistent state once a given number of previously available nodes disappear. We define as consistent state the fact that all remaining nodes are in a normal state concerning the cluster and the available data in the Nodes distributed registry is replicated among remaining nodes. It is paramount to underline that at this stage we are not considering migration processes of workloads in execution in this Edge node, instead we solely focus on the Node management processes. The analysis of recovery times from node churn has included the behavior for 2, 4, 6 and 8 nodes suddenly abandoning the Ad-hoc Edge Infrastructure. These represent percentages of 25%, 50%, 75% and 100% (the disappearance of the cluster). Fig. 4 shows the obtained results for Ethernet and WLAN connectivity, we view this recovery time as the time necessary for the cluster to be in a consistent state once certain percentage of the nodes of the cluster leave the system. Following a logical correspondence with the rest of results gathered in this evaluation recovery times increments as the number of nodes disappearing from the system increase. Both from Ethernet and WLAN (Fig. 4) tests it is apparent that the larger the number of abandoning members of the cluster, the

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

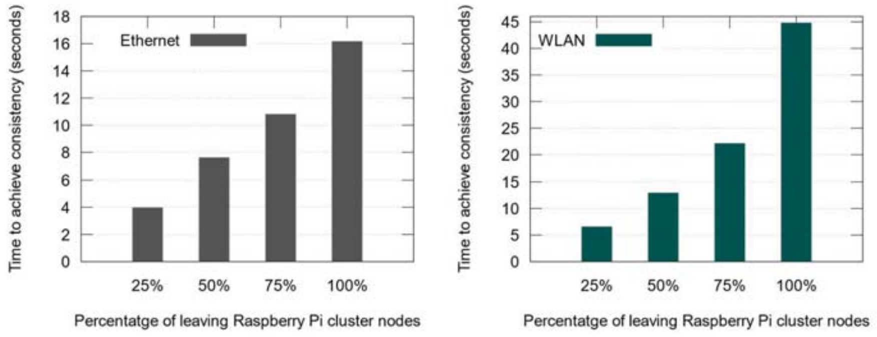

Fig. 4 Time to recover from % node churn in 8 RPI Cluster

longer times to recover. This is determined in the extreme cases which possess more that 75% of node churn, obtaining substantially long recovery times for the two last members remaining in the system

B. Large Scale Evaluation in AWS EC2

Scalability Experimentation

Experimentation in the AWS simulated environment centered on analyzing behavior of two distributed storage systems Apache Cassandra and Etcd in dynamic environments and at scale. The first experiment is based in the latter by reproducing the process of creating from scratch a complete Ad-hoc Edge Computing Infrastructure cluster composed by 10, 20, 30, 40, 50 and 100 nodes, and response times obtained in seconds for this operation. Hence, it is immediately noticeable that creation times of Etcd clusters enhance results obtained by Apache Cassandra both in terms of time and necessary resources. In Etcd we obtain orders of magnitude of less than 1,4 minutes (87 seconds) to set-up a 100 nodes cluster, while necessary time in the same environment for a 10 nodes cluster for Apache Cassandra is 8,5 minutes. When checking resource consumption of the created nodes, the results are aligned: the average memory consumption of an Apache Cassandra node in a 10 nodes cluster is 328 Mb while results obtained for Etcd are 14Mb. Growth on cluster size follows a similar trend obtaining average memory usage of 100 nodes Etcd clusters inferior to 10 nodes cluster for Apache Cassandra. It has to be mentioned that tests for Apache Cassandra were ceased after the creation of 40 nodes, as the cluster state remained unstable.

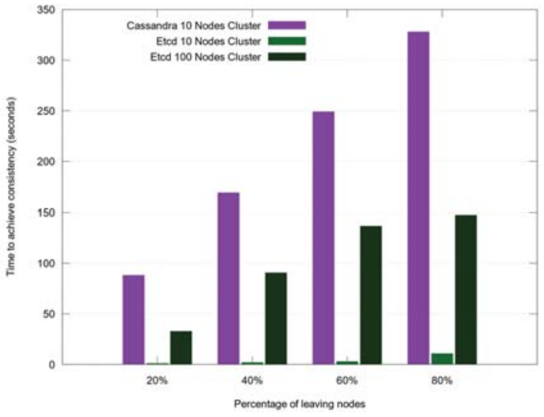

Fig. 5 Time to recover from % node churn in 10 and 100 nodes cluster

It is essential to establish the fact that while the memory consumption for Apache Cassandra is above 280 Mb for a 10-node cluster, it remains steady on similar orders of magnitude as the cluster grows. Conversely, Etcd average memory usage per node increases with the size of the cluster. This is, by all means, explained due to the node synchronization processes among data storage cluster nodes, however it raises an important aspect to take into account in future developments for our Ad-hoc Edge Cloud framework scalability. It is significant to remark that in Etcd the limit on memory consumption of our resource management tools stated in 32Mb to be usable in constrained Edge devices such as Raspberry Pis is in this environment achieved at a cluster size of 30 nodes.

Availability/ Churn rates Experimentation

This experiment employs a 10-nodes cluster and checks the time required to achieve the consistency state under diverse node churn rates: 20% churn rate representing 2 nodes becoming abruptly not available; 40% churn rate with 4 nodes abandoning; 60% with 6 nodes; and 80% corresponding to 8 nodes. The data obtained is illustrated in Fig. 5. It shows that the time to recover, due to the data synchronization processes, is linearly related with the number of nodes withdrawing from the system. As exemplary value, for Cassandra it takes 1,4 min for a 10 nodes cluster to recover from 20% nodes churn rate. Suffice it to say that this process is significantly more performant in Etcd, for which we have experimented 20%, 40%, 60% and 80% churn rates over 10 and 100 nodes cluster obtaining recovery times of 1 second for 20% churn rate in with cluster size of 10 nodes and 15 seconds for 100 nodes cluster. It is remarkable that Etcd is showing notably better recovery times for 80% churn rates than Cassandra for a 10 nodes cluster.

V. Background

The use of constrained and mobile devices as infrastructure resources has been explored -with the help of a number of research areas [8]. For instance, in Fog Computing context, OpenFog Consortium Reference architecture [22] provides a hierarchical view of Fog (edge) comprising IoT devices, Edge and Cloud resources. OpenFog architecture makes the consideration that IoT devices are static and uniquely act as data sources. At the level of Edge it exclusively recognizes its capacity to provide an intermediate comping layer. In [23] Edge cooperation mechanisms (as the ones applied in our work by the Ad-hoc formation of Edge infrastructures) were identified as a

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.

specific requirement for resource management in Fog computing. In recent times, [24] has defined the concept of Mobile Ad-hoc Cloud Computing, which explores the use of mobile devices as a mechanism to "augment various mobile devices in terms of computing intensive tasks execution by leveraging heterogeneous resources of available devices in the local vicinity". Our work develops on top of this concept extending on specific issues to be addressed in this context node churn scalability and heterogeneity.

Focusing on specific works: Hyrax [25] builds a map / reduce Hadoop infrastructure on top of mobile devices equipped with Android operating system. It does not consider mobile device availability issues. Huerta-Canepa developed in [26] the concept of virtual mobile cloud computer providers. in which mobile devices participating in this infrastructure form on a virtual mobile cloud provider. This is conceptually similar to our approach to Ad-hoc Edge Cloud however not providing fault tolerance mechanisms while managing services in the mobile cloud. More recently, diverse works have examined the use of Unmanned Aerial Vehicles (UAVs) in related fields to our work. Valentino in [27] proposes a schema for applying task offloading techniques among drones' clusters. It makes the consideration of task off-loading in a traditional MCC approach, not elaborating in specific characteristics of considering UAVs as Edge resources taking part on ad-hoc formed computing infrastructure.

VI. Conclusion and Future Work

The framework presented in this paper leverages in a framework which takes advantage of the growing compute capacity at the Edge of the network in complex IoT devices, by dynamically forming out clusters of these devices in a decentralized and distributed manner. These Edge devices present specific characteristics in terms of scale, heterogeneity and resource volatility that this framework addresses. In addition, we have validated two critical aspects for resource management this framework: scalability and impact of node churn. Experimentation evidenced the relevance of scale both in terms of having the ability to support churn rates along with the management performance overheads it brings. The next steps in this work will entail extending the mechanisms for admission control and service management in this heterogeneous resource context.

References

[1] AWS Greengrass, https://aws.amazon.com/greengrass/

[2] Azure IoT Edge, https://azure.microsoft.com/en-us/services/iot-edge/

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. *Proc. first Ed. MCC Work. Mob. cloud Comput.* (2012), 13–16. DOI:http://dx.doi.org/10.1145/2342509.2342513´

[4] IDC FutureScape: Worldwide IT Industry 2019 Predictions, https://www.idc.com/getdoc.jsp?containerId=US44403818

[5] NVIDIA Jetson Systems, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

[6] Intel Aero Compute Board, https://software.intel.com/en-us/aero/compute-board

[7] Artificial Intelligence Edge Device Shipments to Reach 2.6 Billion Units Annually by 2025, https://www.tractica.com/newsroom/press-releases/artificial-intelligence-edge-device-shipments-to-reach-2-6-billion-units-annually-by-2025/

[8] Ana Juan Ferrer, Joan Manuel Marquès, and Josep Jorba. 2019. Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. ACM Comput. Surv. 51, 6, Article 111 (January 2019), 36 pages. DOI: https://doi.org/10.1145/3243929

[9] The Economist. 2016. After Moore's law, The future of computing. (2016). https://www.economist.com/News/Leaders/21694528-Era-Predictable-Improvement-Computer-Hardware-Ending-What-Comes-Next-Future .

[10] Docker, https://www.docker.com/what-docker

[11] Matt Richardson, 2016, Docker comes to Raspberry pi, https://www.raspberrypi.org/blog/docker-comes-to-raspberry-pi/

[12] Ros.org, Getting started with ROS and Docker, http://wiki.ros.org/docker/Tutorials/Docker

[13] Container Journal Cisco to Run Containers at the Network Edge, https://containerjournal.com/2017/06/29/cisco-run-containers-network-edge/

[14] Unikernels, http://unikernel.org/

[15] Kata Containers: https://katacontainers.io/

[16] gVisor, https://github.com/google/gvisor

[17] Etcd https://coreos.com/etcd/

[18] Kubernetes, https://kubernetes.io/

[19] Amazon Web Services, https://aws.amazon.com/

[20] Apache Cassandra, http://cassandra.apache.org/

[21] Raspbian, https://downloads.raspberrypi.org/raspbian_lite_latest

[22] The OpenFog Consortium ArchitectureWorking Group. 2016. Architecture Overview. ttps://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf.

[23] Luis M. Vaquero and Luis Rodero-Merino. 2014. Finding your Way in the Fog. *ACM SIGCOMM Comput. Commun. Rev.* 44, 5 (2014), 27–32. DOI:http://dx.doi.org/10.1145/2677046.2677052

[24] Ibrar Yaqoob, Ejaz Ahmed, Abdullah Gani, Salimah Mokhtar, and Muhammad Imran. 2016. Mobile ad hoc cloud : A survey. , July (2016), 2572–2589. DOI:http://dx.doi.org/10.1002/wcm

[25] E. E. Marinelli, "Hyrax: Cloud Computing on Mobile Devices using MapReduce," Sep. 2009. Master's thesis, Computer Science Departmente, Carnegie Mellon University.

[26] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," Proc. 1st ACM Work. Mob. Cloud Comput. Serv. Soc. Networks Beyond - MCS '10, pp. 1–5, 2010.

[27] Valentino, R., Jung, W., & Ko, Y. (2018). Opportunistic Computational Offloading System for Clusters of Drones, 303–306.

Authorized licensed use limited to: UNIVERSITY OF MICHIGAN - DEARBORN. Downloaded on March 18,2023 at 14:08:58 UTC from IEEE Xplore. Restrictions apply.