International Journal of Combinatorial Optimization Problems and Informatics, 12(2), May-June 2021, 1-6. ISSN: 2007-1558.

www.editada.org

Editorial: From traditional Information Technology to Serverless. A novel approach to monitor mobile phones using concepts and principles of Industrial Internet of Things

Roberto Contreras-Masse [1], Alberto Ochoa-Zezzatti [1], Mayra Elizondo-Cortés [2], Vianey Torres [1], Vicente García Jiménez [1], José Mejía [1]

[1] Universidad Autónoma de Ciudad Juárez, Chih 32500, México.
[2] Universidad Autónoma Nacional de México, CDMX, México.

Abstract. The incorporation of smartphones and smart devices in daily business activities and processes is widespread in several industries. Due to its high use, it is required to monitor the device's health and uses patterns to achieve business continuity and conduct code compliance. The monitoring activity is currently implemented in the traditional IT paradigm, either on premises or in the cloud, deploying the solution in servers with fixed capacity and cost involved. This research proposes a novel approach to changing the paradigm by considering smart devices as one live entity with many health metrics ready to exploit their telemetry for monitoring IIoT concepts. The proposed approach introduces the serverless computing paradigm in the cloud to implement IIoT telemetry gather activity at lower costs with scalable capacity, and pro- poses the new architecture that can be implemented by different enterprises in any industry where smartphones and devices are part of daily business activities.
Keywords: IIoT · Cloud Computing · Serverless computing

1 Introduction

Information Technology has been the subject of constant change and evolution. One of the most recent trends is cloud computing, where computing power, storage, network and security are provisioned and accessed through Internet. Cloud computing has also evolved offering different delivery methods, such as Infrastructure as a Service (IaaS), Platform as a service (PaaS), and Software as a Service (SaaS). All three-delivery models are offered under a shared responsibility schema to enterprises and individuals. There are groups of IT managers devoted to traditional Information Technology (IT), and solve modern problems with traditional elements, i.e. IT on-premises. Other groups of IT experts are willing to solve the same problem with cloud computing. There are differences, advantages and disadvantages depending on what architecture is designed. This article provides a novel alternative to the traditional approach currently implemented within a multinational company, to solve the challenges to monitoring mobile devices provided to more than 10,000 workers. Sections organize this document. Related work analysis and alternatives are discussed in section two. Methodology, problem description along with our approach is presented in section three. Section 4 discusses the benefits and disadvantages found and evaluated, and finally, fifth section presents conclusions and further research.

2 Related research

The use and popularity of mobile devices (smartphones, tablets, PDA's) is obvious. There are more than 3.5 billion smartphones deployed around the globe, according to Statista in their results of 2019. Traditional handhelds are being substituted with smartphones in different industries, to provide more functionality and flexibility to drivers. As this is a critical working tool, it requires to be monitored constantly. Currently, there are software products that extract smartphone's health check, such as AirWatch, which implements mobile device management. Usually, this kind of tool requires installing agents on devices that communicate to central infrastructure, storing data for data insight purposes, and providing a management user interface [2]. The central infrastructure requires at least two servers, one to process the incoming information from device and second server to host the database. This configuration can work for one or two thousand devices, and has no analytic capacity. To add this last capacity, it requires another two servers, one to extract, transform and load data into the second server acting as a data warehouse. It can be

seen immediately the increase quantity of servers in a traditional IT environment. To know a device health and usage, there are several key elements to look for, with multiple data derived from each component status and registries within the device. Frederick Neumeyer patented a method to monitor battery life [5] reading memory, battery life counter and records of energization and operations tables. Just by looking at battery information, anyone can corroborate the information is tracked in real time looking at their smartphone screen.
Therefore, data size is very small (just record a number between 0-1) but recorded at high rate to achieve what people perceive as real-time. The volume of data generated can be assumed to be very large. Therefore, the traditional IT infrastructure needs to scale as enterprises adds more devices. To achieve scalability, cloud computing provides a good option in IaaS delivery, however, this delivery method translates to fixed cost and not very flexible to new paradigms, such as microservices. Serverless computing emerges as the compelling paradigm to build microservices-based applications native in cloud environments [1]. In a nutshell, serverless reduces the control on hardware and in- crease the focus on optimized software developed in different languages [8]. This is a very good option to handle increasing data volume, as the cloud provider manages the underlying resources.

The high volume of data transmitted in very small packages is a problem present in Industrial Internet of Things (IIoT), where this data produced by de- vices is called telemetry. To speed up deployment of solutions to gather telemetry, different authors have included in their proposals the use of managed services to collect telemetry. Tarnberg (2016) uses AWS IoT to gather telemetry from a smart traffic control system [6]. Martinov (2020) uses Azure IoT Hub Center as the central component to gather JSON and XML from several heterogeneous devices [4]. Trilles (2020) opted for open-source products in his experiment, however explored serverless in good detail [7]. AWS IoT and Azure IoT Hub are options available, including Google Cloud Platform IoT. These options are being considered and selected by multi-criteria decision-making methodologies [3], but it has not been reported as a method for smartphone telemetry gathering.

3 Methodology

The current solution in place is built with monitoring agents installed in smartphones sending telemetry each minute. Figure 1 shows the traditional IT architecture, where consist of several components in the back end to process this amount of data. The information gathered is: 1) Battery related (percentage charged, charging or draining, and temperature); 2) Connectivity related (Wi-Fi reception, GSM reception, GSM Band, talk time accumulated, GPS coordinates); and 3) Memory and storage-related (RAM used and available, internal storage used and available). Although it is constructed to be flexible, it requires acquiring hardware, install it, and manage it, every time escalation is needed. Although works, current architecture has limited capacity and could serve only a certain number of devices before reaching the saturation point. To find the current limit, a performance simulation is executed to ramp up number of devices sending information to receive message servers (RMS). Figure 2 shows CPU usage (percentage) as the number of devices increases. It identifies the thresholds providing signs of alerts. In this simulation, the thresholds are set for: 65% means a warning zone; 80% is the alert zone where the performance of CPU starts to being compromised; above 95% is a high risk of CPU saturation and it is never recommended to reach that level in production. Last, the simulation calculated the number of devices pushing the CPU beyond 100% and it means the maximum capacity the CPU can handle. The simulation depicts a curve corresponding to sigmoid function as

$$f(x) = \frac{\lambda}{1 + e^{-\beta(\mu - x_i)}} \qquad (1)$$

3.1 Calculate expenditure in Traditional IT

From simulation was found the range of devices per server as [981, 1141], so it can be calculated a midpoint of 1061 devices per server. In other words, this ´platform server of same characteristics will be able to handle 1061 devices. This finding helps to calculate the number of servers required to fulfill demand. It is common practice to roll out a new application to workforce in stages, starting with a pilot and then keep growing according to strategy, time and budget. Figure 3 shows the classic problem enterprise face when dealing with server provisioning. As shown in this study case, both graphs show under capacity and over capacity at certain months. For example, during the first two months, the pilot is planned in 500 devices, and then jumps to 1350 devices in month two. There is overcapacity in month 1, but immediately in month 2 there will be a under capacity. Analyze cost (not considering networking elements such as firewalls, load balancers and licensing) to compare against serverless is required. Therefore, Table 1 shows the month-by-month expense. An average server cost USD 4,000 plus USD 1,000 of set-up one time, giving a total investment of 50,000 USD during the first 9 months, or 5,555.55 per month.

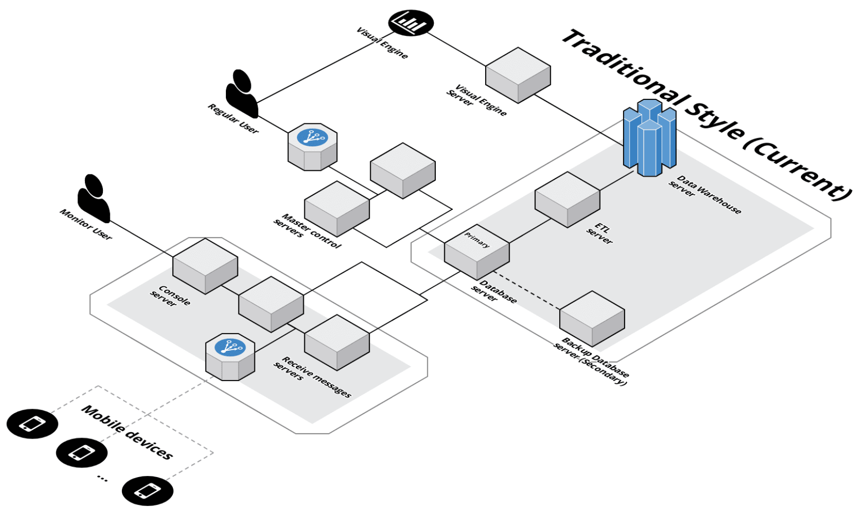

Fig. 1. Traditional architecture as it is now.

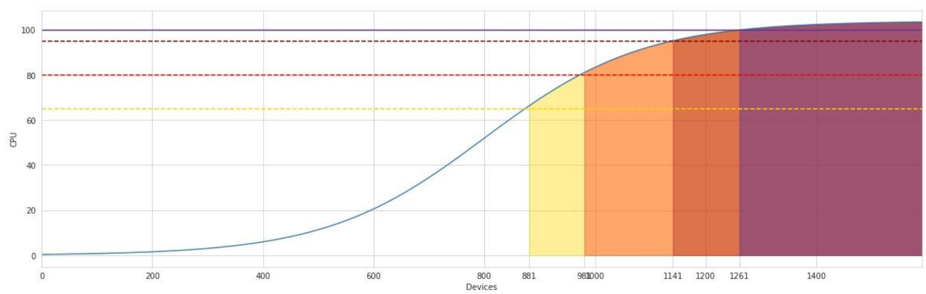

Fig. 2. CPU usage for a single server under workload analysis for simultaneous devices being monitored. (Source: own creation)

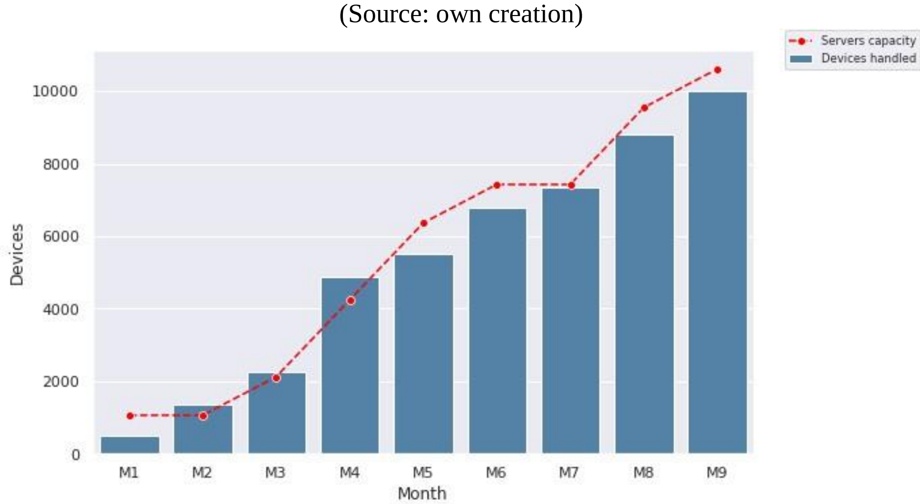

Fig. 3. Server growth related to device roll out by month. (Source: own creation)

Table 1. Expenses per month

M1	M2	M3	M4	M5	M6	M7	M8	M9	Total
5,000		5,000	10,000	10,000	5,000		10,000	5,000	50,000

The proposed architecture is depicted in Figure 4 uses the IoT services pro- vided by AWS or Azure. For this study case and to be able to compare against the cost per month, it is required to calculate the messages that will be sent by device during a time frame of 16 hours of monitoring (two regular shifts). The number of messages can be calculated by

$$M_i = \varphi * m * d \qquad (2)$$

where Mi are messages per device, φ is the frequency of telemetry messages per hour, m is hours where device is being monitored, and d is the average number of days per month. For this paper, φ = 60, m = 16, and d = 26, for a total of Mi = 24,960 messages. Each message is 202 bytes, therefore, the data transmitted per device will be 4.8 MB. Taking as an example AWS IoT Core service and Azure IoT Hub as an option to eliminate servers to receive data, the only possibility to compare both services against traditional IT is in cost. Thus, the architecture proposed evaluates those two options.

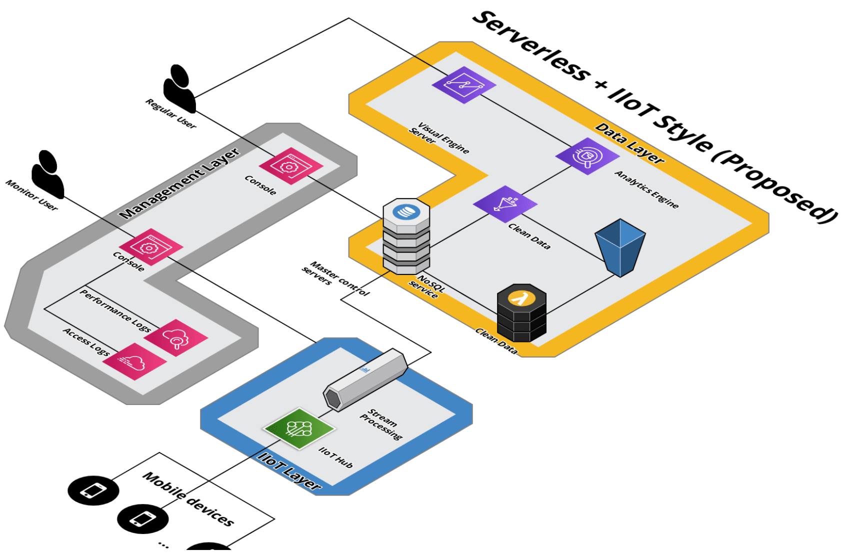

Fig. 4. Serverless architecture proposed using IIoT approach and serverless compo- nents.

3.2 AWS IoT Core

AWS IoT core documentation shows three components of the pricing: 1) connectivity, calculates minutes of connection, 2) messaging, based on number of messages and size, 3) Registry devices by number of operations, 4) rules. All services are charged in millions. For simplicity, this exercise uses connectivity, messaging and registry features. They are calculated as:

– Connectivity cost is calculated as minutes of connection * service cost (0.08)
/ 1M connections.
– Messaging cost is calculated as messages * service cost (1) / 1M
– Registry cost is calculated as operations * service cost (1.25) / 1M

3.3 Azure IoT Hub

Azure provides IoT hub with a single pricing component 4, the total number of messages per day. The edition to choose depends on this metric. Therefore, it will be required to calculate the total number of daily messages produced by the number of devices rolled out by month.

4 Results and discussion

The calculation of estimated expenses in both IoT platforms are presented in a side-by-side comparison in Table 2.

Table 2. Expenses per month

Service	M1	M2	M3	M4	M5	M6	M7	M8	M9	Total
Devices	500	1350	2250	4870	5500	6800	7360	8800	0	10000
IT	5,000	0	5,000	10,000	10,000	5,000	0	10,000	5,000	50,000.00
AWS	13.49	36.44	60.73	131.44	148.44	183.53	198.64	237.51	269.89	1,280.10
Azure	250	250	250	250	250	500	500	500	500	3,250.00

Comparing the options in the common metric, US Dollars, it can be clearly seen the traditional IT total during the nine months of roll out project the amount of $50,000 expended up front. Usually, lifetime of IT servers are three years, so it is valid to divide that amount in three years for a total of $16,667.66 per year. In any serverless IoT hub analyzed, the total cost per year is: 1) AWS for $2,098.78, calculated by adding 1,280.10 plus three months of 269.89 to complete the year; 2) Azure, in the other hand, yearly cost estimated is 4,750 following same logic to complete one year. It is clear any IoT Hub options is better in cost efficiency. The other important aspect is the flexibility to scale out fulfilling the demand. In traditional IT servers must be acquired and installed, and there will be a waste of resources at some months and under capacity and performance risk in other months. Figure 3 shows clearly month M4 will have more devices signed up that the capacity can handle, and it will be until month M5 where capacity is well over demand, meaning another month of wasted resources. This happens in the following months and finally, at month M10, there will be a small over capacity for any contingency. Along with this analysis, Figure 5 shows AWS as the closest demand fulfillment. Azure shows a huge overcapacity that can be seen as waste, but this is due to its commercial schema. Traditional IT can be seen as the second-best option, but the alternative gets in last place if it is considered the cost.

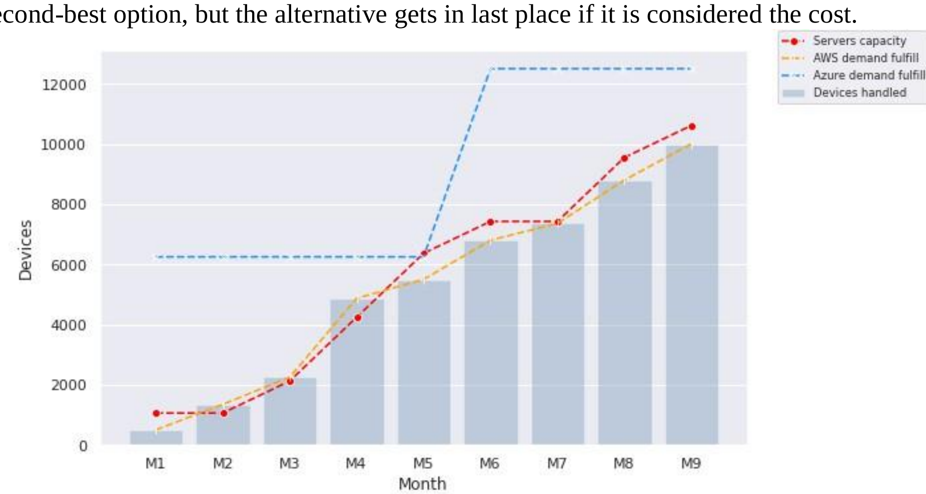

Fig. 5. Comparison capacity fulfillment of different alternatives.

5 Conclusions and Future Research

It can be seen that cost wise, traditional IT is much more expensive compared to serverless computing. The options analyzed were services provided that are full managed and scalable that can easily accommodate a growing number of devices. In addition to this feature, in the event of reducing number of devices, traditional IT cannot be returned or just removed, meaning it will be a great amount of computing resources wasted. Serverless computing and IoT platforms will shrink based on number of messages. In that event, the closer to demand will be the better solution. This feature will help enterprises to save on expenses while providing a good service to their users.

The approach to use an IIoT solution to gather data from devices that are not sensors is novel and can serve. IIoT has been typecast as an industrial environment, where sensors and actuators are the only source of data and telemetry available. Here, smartphones have a variety of elements as good sources of information and can use internet protocols such as http and https to send messages. It is very convenient the IoT platforms are http/https enabled, allowing communication with different devices that implement those protocols. With those pieces put together, it is promising to use IIoT concepts into different devices, such as smartphones or mobile devices.

In future research, it will be focused to review other applications that could leverage the IIoT paradigms of small packages at high rate to gather data and generate insights. In addition, the next study will be to test message ingestion speed to compare if IIoT platforms are an advantage or disadvantage against traditional IT.

6 References

1. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., Suter, P.: Serverless computing: Current trends and open problems. In: Research Advances in Cloud Computing, pp. 1-20. Springer (2017)
2. Batool, H., Masood, A.: Enterprise mobile device management requirements and features. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 109-114. IEEE (2020).
3. Contreras-Masse, R., Ochoa-Zezzatti, A., Garcia, V., Perez-Dominguez, L., Elizondo-Cortés, M.: Implementing a novel use of multicriteria decision analysis to select iiot platforms for smart manufacturing. Symmetry 12(3), 368 (2020)
4. Martinov, G., Kovalev, I., Chervonnova, N.Y.: Development of a platform for collecting information on the operation of technological equipment with the use of industrial Internet of things. In: IOP Conference Series: Materials Science and Engineering. vol. 709, p. 044063. IOP Publishing (2020)
5. Neumeyer, F.C.: Battery life monitor system and method (Jun 16 2020), uS Patent 10,687,150
6. Tarneberg, W., Chandrasekaran, V., Humphrey, M.: Experiences creating a frame- work for smart traffic control using aws iot. In: 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC). pp. 63–69. IEEE (2016)
7. Trilles, S., Gonzalez-P´erez, A., Huerta, J.: An iot platform based on microservices and serverless paradigms for smart farming purposes. Sensors 20(8), 2418 (2020).
8. Van Eyk, E., Toader, L., Talluri, S., Versluis, L., Uta, A., Iosup, A.: Serverless is more: From paas to present cloud computing. IEEE Internet Computing 22(5), 8-17 (2018).