

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316352914>

What Does I(o)T Cost?

Conference Paper · April 2017

DOI: 10.1145/3053600.3053601

CITATIONS

0

READS

25

2 authors, including:



[Attila Kertész](#)

University of Szeged

76 PUBLICATIONS 616 CITATIONS

SEE PROFILE

All content following this page was uploaded by [Attila Kertész](#) on 02 May 2017.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

What Does I(o)T Cost?

Edua Eszter Kalmar

University of Szeged

Dugonics ter 13.

Szeged, Hungary 6720

kalmar.edua.eszter@stud.u-szeged.hu

Attila Kertes

University of Szeged

Dugonics ter 13.

Szeged, Hungary 6720

keratt@inf.u-szeged.hu

ABSTRACT

Though in recent years cloud-based solutions already started to dominate the Internet of Services, with the appearance of things of the Internet of Things (IoT) paradigm more complex systems have been formed that still need a significant amount of research. The large amount of data produced by these devices requires cloud services to be efficiently processed and meaningfully visualized. In this paper we perform cost-based investigations of two popular IoT application groups to help users to better understand IoT cloud ecosystems. We also compare existing IoT cloud providers by estimating on-demand service costs considering different circumstances, diverse messages sizes, number of messages, frequency of message sending and different virtual machine or application service configurations. We also implemented the defined scenarios with the IBM Bluemix IoT Platform and measured the data usage and prices.

KEYWORDS

cloud computing, internet of things, cost estimation

1 INTRODUCTION

In the recent years cloud-based solutions started to dominate the Internet of Services, with the appearance of the Internet of Things (IoT) paradigm more complex systems have been formed that still need a significant amount of research. Things can interact and communicate among themselves and with the environment by exchanging data and information sensed, and react autonomously to events and influence them by triggering actions with or without direct human intervention. Recent trends and estimations call for an ecosystem that provides means to interconnect and control these devices with cloud-based solutions. The large amount of data produced by IoT devices requires cloud services needs to be efficiently processed and meaningfully visualized. With the help of cloud solutions, user data can be stored in a remote location, and can be accessed from anywhere. There are more and more PaaS cloud providers offering IoT specific services (e.g. Amazon AWS IoT Platform, Azure IoT Suite). Some of these IoT features are unique, but every PaaS provider addressing IoT has the basic capability to connect to and store data from devices.

In this paper we perform cost-based investigations for concrete IoT use cases by two scenarios, in which we compare existing, popular IoT cloud providers by estimating on-demand service costs considering different circumstances, diverse messages sizes, number of messages, frequency of

message sending and different virtual machine or application service configurations. We also implemented the defined scenarios with IBM Bluemix IoT Platform and measured the data usage and prices.

The remainder of this paper is as follows: Section 2 introduces related approaches in the field of IoT Clouds. Section 3 presents example scenarios of typical IoT Cloud systems, and details our methods to estimate resource usage costs. Section 5 presents real cost usage validations for a concrete provider, and the contributions are summarized in Section 6.

2 RELATED WORKS

The integration of IoT and clouds has been envisioned by Botta et al. [1] by summarizing their main properties, features, underlying technologies, and open issues. A solution for merging IoT and clouds is proposed by Nastic et al. [2]. They argue that system designers and operations managers face numerous challenges to realize IoT cloud systems in practice, due to the complexity and diversity of their requirements in terms of IoT resources consumption, customization and runtime governance. They propose a novel approach to IoT cloud that encapsulates fine-grained IoT resources and capabilities in well-defined APIs in order to provide a unified view on accessing, configuring and operating IoT cloud systems, and demonstrate the framework for managing electric fleet vehicles.

Atzori et al. [3] examined IoT systems in a survey. They identified many application scenarios, and classified them to five application domains: transportation and logistics, health-care, smart environments (home, office, plant), personal and social, finally futuristic domains. They described these domains in detail, and defined open issues and challenges to all of them. Concerning privacy, they stated that a lot of information about a person can be collected without the person being aware, and control on all such information is impossible with current techniques.

Based on these works we derived two IoT Cloud scenarios and investigated their operational costs at different providers. The following sections define our model of cost calculations based on publicly available pricing information, and validate our calculations with real experiments.

3 CALCULATING IOT CLOUD OPERATION COSTS

In this section, we present cost-based comparison of concrete IoT use cases represented by two specific scenarios, in which we compare existing, popular IoT cloud providers

by estimating on-demand service costs considering different circumstances. We considered the following, most popular IoT cloud providers: (i) Microsoft and its IoT platform called Azure IoT Hub [4], (ii) IBMs Bluemix IoT platform [6] is the second we surveyed, last but not least we studied the services of (iii) Amazon (AWS IoT) [8] and finally, (iv) Oracles IoT platform [7]. We took into account the prices publicly available on the websites of the providers and when we found it necessary we asked for further information or clarifications via email from the providers. The calculation of the prices depends on different methods. Some providers bill only according to the number of messages sent, while others also charge for the number of devices used. The situation is very similar if we consider the virtual machine renting or application service prices. One can be charged after GB-hour (uptime) or according to a fix monthly service price. This price also depends on the configuration of the virtual machine or the selected application service, especially the amount of RAM used or the number of CPU cores or their clock signal.

In our model we define two use cases for cost estimations based on real-world examples that vary in the following parameters: total number of sent messages in a certain period of time, the number of devices used, and the capacity of the virtual machine used to provide gateway services. The first case is a system where people are examined with the help of wearable smart devices (smart watches) and pieces of medical data are sent to the application running in the cloud of the provider for further processing. Secondly, we considered a use case of a traffic light control system in a smart city. This scenario is significantly bigger in the number of devices used and messages sent than the previous one. It also requires a stronger virtual machine or application service for smooth sensor data processing. We estimated how our application would be charged after a whole month of uptime running in the cloud of the providers mentioned before. In our model, the total cost of executing an application consists of two price categories: (1) IoT and device prices and (2) cloud side prices. In case (1), we may be charged after the tier (a package) used or only after the resources used. The latter is also called "pay as you go" billing method, it means that we only pay for what we really use. At some particular providers, we need to pay for both of these two methods. Moreover, there are message prices as well. If we pay for a tier (if it is possible at the particular provider) then the price of a message is not so important because the tier includes prices of a fix number of messages. However, the price of the tier depends on the number of messages we want to send; more messages are covered by bigger tiers cost more money. If we use a provider with a pay as you go category, then the price of a message becomes more important. In some cases, we are charged after data exchanged not the number of messages sent but the data used can also be covered by a tier. Finally, it may occur that we need to pay for the number of devices used. To run an IoT application we also need to pay for a virtual machine or application/compute service or runtime to operate a gateway service – covered by case (2). There

can be a fix monthly price for a service but GB-hour price can be charged as well.

Azure IoT Hub [4] charges us after the chosen edition/tier. This means that there are intervals for the number of messages used in a month. Azure also comes with some extras when we start to use its services, as well as some of the providers do so, but we do not take extras into consideration because we investigate general situations. There is a restriction for message sizes which depends on the chosen tiers. One can choose from four tiers, Free, S1, S2, S3. Each of them vary in price and the total messages allowed per day. Message size of the Free tier also differs from the other tiers. In the Free edition, devices can only send a lot smaller messages than in the other editions. Regarding to the cloud side prices we need to count with an application service price and there is no GB-hour price because the service is in full uptime. We have the opportunity to choose from a wide variety of configurations, selecting the number of processor cores, RAM used and storage capacity, affecting the price of the application service. IBM Bluemix IoT platforms pricing method follows completely the "pay as you go" method, and it can be seen in Bluemix pricing sheet under the Internet of Things section and at Internet of Things platform [6]. Bluemix only charges after the MegaBytes exchanged. We differentiate three categories and each of them comes with a different price per MB. There are three categories for the data used in MegaBytes (MB) and each category has its own price per the MBs exchanged. The more MBs we use and thus select a bigger category, the less price per MB we get. Working with Bluemix we need to pay for the runtime as well to run our applications. It is configurable, depends on the number of instances and the RAM used, and has a fix monthly price. On the top of that, we will be charged for GB-hour price, too. Amazons IoT platform can also be classified as a "pay as you go" service. Its billing method [8] works out incredibly easily. Prices are based on publishing cost (the number of messages published to AWS IoT) and delivery cost (the number of messages delivered by AWS IoT to devices or applications). A message is a 512-byte block of data and the pricing in EU and US regions denotes \$5 per million messages. In addition, there is no charge for deliveries to some other AWS Services. So, there is only price per message billing which can be affected by the size of messages because there is a limit for message size. Using Amazons IoT solution we also need a virtual machine for the gateway service. We can choose from a wide range of virtual machine configuration affecting its price and GB-hour price will be charged as well. In our calculations 1 USD converts to 0.914039185 Euros.

Finally, we investigated how prices can be calculated at Oracles IoT solution. The pricing method is slightly different from the three providers described before. We can say that its rather similar to Azures tiers than a completely "pay as you go" billing like in Bluemix. The information was gathered from [7] and we calculated with the so-called Metered Services. There are four product type categories regarding the used devices (wearable, consumer, telematic,

commercial/industrial). Each category type has a price per used device type. The four device/product type category determine the monthly device price and the number of messages that can be sent by that particular type of device. In addition, there is a restriction on how many messages can a particular type of device deliver per month. In case, the number of messages sent by a device is more than the devices category permits, an additional price will be charged according to a predefined price per thousand of messages. Concerning the cloud side, in Oracle we should also pay for a compute service and daily uptime of our application. The number of CPU cores also affect the price of this service.

Concerning cloud-based cost requirements, for the first scenario, we need about 1 GB of RAM and 1 CPU core. For the second scenario, we estimated that about 2-4 GB of RAM and 2 CPU cores could run our application smoothly, and finally we decided to take more than 4 GB of RAM and 4 CPU cores for the third scenario. The exact parameters for these providers can vary from the configurations mentioned just before.

We also collected pricing information for gateway services from the providers official sites. The pricing of Azures application service can be found at [9], Bluemix runtime is in its pricing sheet under the Runtimes section [6], Amazon EC2 On-Demand prices are described at [10] and we can find the pricing of Oracles compute service at [11]. We used the prices of the Metered Services. By clicking on the Buy Now button next to Metered Services sign we can navigate to a detailed pricing calculator [12].

4 COMPARISON OF IOT CLOUD OPERATIONAL COSTS OFFERED BY FOUR PROVIDERS

Scenario I: Examining People With Smart Watches To Acquire Medical Data

Our first scenario representing a large group of IoT applications is a use case where 10 people are examined with the help of wearable smart devices. In this case 10 smart watches monitor and send pieces of medical data about the patients regularly to a gateway application running in the cloud of a provider for further processing. The detailed information about the number of messages, devices and uptime hours is shown in Figure 1. We expect our application to run for a whole month (31 days), that means 744 hours. The considered messages size is 0.7 Kilobyte (KB) in this scenario. It might sound a little big for just sending some medical data but we would like to measure many things about the patients. We decided that every device wore by the people should send one message in every hour. This means 24 messages a day per device and $24 * 31 = 744$ messages per month per device. Total messages per day works out by multiplying the messages per day per device by the number of devices, hence $24 * 10 = 240$ messages are sent per day by all the devices. Since we already got sum of the messages sent in a month per one device, we can easily say that the number of the total messages in a month is the previous number (744)

multiplied by the number of devices, so it is 7440 in the case of the first scenario. The last thing we need to determine for the estimations is the amount of data exchanged monthly. To do so, we just multiply the number of total messages in a month by the message size and then divide the number we got with 1024 to get the result in MegaBytes and not in KiloBytes. The result is 5.086 MB.

| | |
|--------------------------------------|----------|
| Devices | 10 |
| Device type | Wearable |
| Message size | 0.7 |
| Messages / month / device | 744 |
| Total messages / day | 240 |
| Total messages / month | 7 440 |
| MB exchanged / month | 5.086 |
| Messages transferred / device / hour | 1 |
| Test duration (days) | 31 |
| Full uptime in months (hours) | 744 |

| | |
|--------------------------------------|------------|
| Devices | 128 |
| Device type | Telematic |
| Message size | 0.05 |
| Messages / month / device | 133 920 |
| Total messages / day | 552 960 |
| Total messages / month | 17 141 760 |
| MB exchanged / month | 837 |
| Messages transferred / device / hour | 180 |
| Test duration (days) | 31 |
| Full uptime in months (hours) | 744 |

Figure 1: Basic configuration information for Scenario I and II

Working with Azure has the highest benefit in this scenario because it is free as displayed in Figure 2. In Azure, there is a size restriction regarding the messages sent. It is 0.5 KB if the number of total messages per day is less than 8000. We could assume that the previous condition is true since we have 240 total messages a day if we do not consider the size limitation. After considering the limitation, we get 480 messages because we need to divide the given message size with the size limit and then round up the result, and just then multiply this result by the number of total messages a day. So, we have 480 messages a day and that is definitely less than 8000, that is why we calculate with no price in this case. This scenario goes into the Free tier with 0 price. At the cloud side the price of the application service is also 0, since Free F1 tier is a free service of Azure. In summary, Azure is a good choice for the first scenario when we would like to build a small system with only a few devices and not many messages per day. It is the less expensive, actually free, from the four providers surveyed for this use case.

Bluemix provides a quite more expensive service, it costs about 57.3 Euros all together. Since we have 5.086 MB exchanged per month as shown in Figure 2, we pay 0.00097 Euro per each MB. So, that means $0.00097 * 5.086$ which

| | Azure | Prices | Bluemix | Prices | Amazon | Prices | Oracle | Prices |
|---|-------|----------|---------|--------------------|--------------|--------------------|--------|----------------|
| IoT fix prices and device side | | | | | | | | |
| Device price / month | - | | - | | - | | + | 4.6015 |
| "Price / message" pricing | - | | - | | + | 0.06800451536 | + | |
| "X messages / month" pricing (tears) | + | 0 | - | | - | | + | 0 |
| Data exchanged (in MegaBytes) | - | | + | 0.004933359375 | - | | - | |
| Message size limit | 0.5 | | - | | 0.512 | | - | |
| Total messages / day after size limit check | 480 | | | | 480 | | | |
| Cloud side | | | | | | | | |
| Instance price | + | 0 | + | 18.15 | - | | + | 75 |
| GB-hour price | - | | 0.0526 | 39.1344 | 0.0118825094 | 8.840586994 | (+) | |
| TOTAL PRICE / MONTH | | 0 | | 57.28933336 | | 8.908591509 | | 79.6015 |

Figure 2: Cost estimation for Scenario I

equals 0.004933359375 Euro. This was the IoT price part of the final sum. The other part is the cloud side. We need to pay 18.15 Euros for the runtime which can be read in Figure 2. Moreover, there is a GB-hour price as well, it is 39.1244 by multiplying the uptime hours (744) by the 0.0526 GB-hour price. This last three numbers result the total price, that we should pay for this scenario with the given circumstances, which is that almost 57.3 Euros for a month. This is more expensive than Azure but it is cheaper than Oracle for example. We would say this is not the best provider for our first scenario. Maybe with different parameters it would be worth choosing it.

Amazon comes with a lower price compared to Oracle, it would make us pay only ~ 9 Euros for almost the same services. It charges us a price for each message exchanged. Amazon uses 512-byte block of data for messages, so we need to do our calculation with 480 messages per day because a 0.7 KB (716.8 byte) payload is billed as two messages. So we multiply 480 by the number of the test duration in days (31) and then with the price per 5 million messages. This is the following: $480 * 30 * 4.570195925 / 1000000$ which equals 0.06800451536. The number 4.570195925 means 5 USD in Euro. So, we can see that 480 messages cost less than 0.1 Euro. We need to add the cloud side price to this number. A GB-hour price can be seen in Figure 2, so we consider the total uptime hours multiplied by the GB-hour price the result which is ~ 8.8 and then add the IoT price we calculated before to get the total price for a whole month. All in all, Amazon is not an expensive provider, what is more, it is actually the cheapest from the four providers (considering more scenarios). We exemplify this statement later in the other scenarios.

The last one is Oracle, the most expensive provider from the four we considered, our first scenario would cost almost 80 Euros for a month according to our estimation. As far as we use smart watches here, the device type is Wearable, so

we calculate with 0.46015 Euro per device per month, and to Wearable product type belong 1500 messages per device per month. Using these numbers, we can state that the number of devices used (10) multiplied by the formerly mentioned device price results the total device price for all our 10 devices which is 4.6015 Euros as shown in Figure 2. It is worth mentioning that neither of our 10 devices send more messages a day than 1500, that is why we do not charge an additional message price. The price of the cloud side is pretty easy to figure out. With the help of Oracles pricing calculator [12] we can come to know this price by selecting 744 uptime hours in the form and it displays the amount of money charged which is 75 Euros for a whole month. So, the price of the compute service depends on the uptime hours. To summarize, Oracle can be classified to the expensive providers category and this will also be confirmed with the following scenarios. The device price makes the total sum a lot higher while it does not take part in other providers pricing.

Scenario II: Traffic Light System of a Smart City

As our second example, we chose a use case of a traffic light control system in a smart city. This scenario is significantly larger than the previous one regarding the number of devices used and messages sent. It also requires a stronger virtual machine or application service to operate a gateway for smooth data processing. We use 128 devices referring to a study and implementation of a smart city in Messina [13].

We can see the detailed information depicted in Figure 1. The amount of the uptime hours is the same as in the first scenario. We would like to run our application for a whole month (744 hours mean 31 days). In this scenario and in the third one we worked with message sizes up to a maximum of 0.05 KBs. Devices send messages in every 20 second which means 180 messages in an hour. From the previously mentioned value we can assume that to get the total number of messages per device for the whole month we need to determine the messages sent by a device during a

| | Azure | Prices | Bluemix | Prices | Amazon | Prices | Oracle | Prices |
|---|---------|---------------|---------|-----------------------|-----------------|-------------------------|--------|-------------------------|
| IoT fix prices and device side | | | | | | | | |
| Device price / month | - | | - | | - | | + | 296.96 |
| "Price / message" pricing | - | | - | | + | 78.19682 2226 | + | |
| "X messages / month" pricing (tears) | + | 421.65 | - | | - | | + | 0.795084 8 |
| Data exchanged (in MegaBytes) | - | | + | 0.81189 | - | | - | |
| Message size limit | 4 | | - | | 0.512 | | - | |
| Total messages / day after size limit check | 552 960 | | | | 552 960 | | | |
| Cloud side | | | | | | | | |
| Instance price | + | 94.11 | + | 112.83 | - | | + | 139 |
| GB-hour price | - | | 0.0526 | 39.1344 | 0.047530 038 | 35.36234 797 | (+) | |
| TOTAL PRICE / MONTH | | 515.76 | | 152.7762 9 | | 113.5591 702 | | 436.7550 848 |

Figure 3: Cost estimation for Scenario II

day (180 * 24) and then multiply it by the number of days (31) while we run this scenario. We get a considerably bigger number than in the first scenario, in fact it is 133920 messages per month per device. The total messages per day come out of the number of the messages sent by a device during a day (180 * 24) multiplied by the number of devices (128), so the result is 552960 which is also a lot bigger than the numbers in the previous scenario. Furthermore, we can count out the total messages per month including all the devices by just easily multiply the number of devices by the number of messages per month per device which means 128 * 133920 = 17141760. We can estimate the exchanged Megabytes if we multiply the number of total messages per month by the message size given in KiloBytes so we then divide with 1024 and then we get the result of 837 MB.

In the second scenario Azure seemed to be really expensive compared to the other providers. Bluemix and Amazon cost less than a half of the price of Azure and Oracle is just a little cheaper than Azure. We use the same methods as in the first scenario to estimate the prices. Since the size of the messages is very low (0.05 KB) we do not need to calculate with double numbers of messages. With that many messages a day we need to pay for the S2 tier which costs 421.65 Euros as shown in Figure 3. and Table 4.2. and the Basic B2 application service of Azure costs 94.11 Euros. The sum of these two numbers is 515.76 which can be said a very high price. As stated before, Azure is very expensive, it is not really worth to use its service with the given circumstances. The next one is Bluemix which is a lot cheaper compared to Azure in this case. We would be charged for a total of ~153 Euros working with Bluemix. The price for the data exchanged is less than 1 Euro, it is 0.81189 Euro. After each of the ~837 MB we need to pay 0.00097 Euro. As for the cloud side, we considered a stronger runtime than in the first scenario so it is of course reflected on the price. Table 4.6. shows that the price of this runtime is 112.83 Euros and the

GB-hour price is 39.1344 because of the 0.0526 GB-hour price and the 744 hours long uptime. We add these three main results and get the total result. To summarize the services offered by Bluemix, it may be a good choice for this scenario, although Amazon is a little cheaper. Amazon with its winner price for this second scenario charges up only about 113.5 Euros according to our estimations. Using the same method when calculating the price for all the messages in the first scenario we can say that the IoT related price is ~78 Euros as displayed in Figure 3. as well. The virtual machine costs ~35 Euros which is determined by the product of the uptime hours and the GB-hour price. Adding up these two partial results, ~78 and ~35, we get the final price for the services of Amazon, which seems to be the best choice.

Finally, we estimate the costs of Oracle again but now for the second scenario. Devices used here are considered to be classified in the Telematic product type. That is why the device price is 2.32 Euros and devices can exchange 100000 messages per month. With this information, we can say that the price of all the devices for a whole month cost the product of the number of devices and the device price, which means 296.96 Euros. However, we need to calculate with other IoT related prices in this case because the number of messages allowed a day with telematic type is 100000 and we exceed this value with the 133920 messages per device per month. 1000 additional messages are charged up at a cost of 0.02344 Euro. This all means that we generated 33920 extra messages that we need to pay additional message price for which is 33920 / 1000 * 0.02344 = 0.7950848 Euro. For the cloud side, we pay 139 Euros for the compute service with the required configuration. To get the final price we add up the device price and the price for the additional messages and the computes price which results in ~437 Euros. All in all, using Oracle we get a nearly as expensive service as Azures. Bluemix and Amazon is far better for this situation according to our estimations.

5 EXPERIMENTAL COST-BASED MEASUREMENTS WITH IBM BLUEMIX

As the next step, we performed experimental measurement of the defined IoT scenarios to confirm our former investigation. To accomplish this goal, we used IBMs Bluemix IoT Platform [5]. After registration for Bluemix, we created a Cloud Foundry Application runtime for a Node.JS application with 1 GB RAM and 1 instance. We also needed an IoT Service to handle the messages between the application and the devices; it lets our application communicate with and consume data collected by our connected devices, sensors and gateways. Finally, we created a device-side program to connect to the IoT service. We developed special scripts to simulate the necessary devices used in the second scenario.

Cost-based Measurements of Scenario II

Considering the second scenario, we use 128 devices and each of them sends messages in every 20 seconds. A sample message of 0.053 KB we used in this case is the following:

```
{ "d" : { "id" : "1uz6", "s" : "1", "t" : "2016 - 12 - 01 09 : 00 : 00" } }
```

It means that the traffic light device (with id 1uz6) shows red sign ($s = 1$) at the given time (t). In this scenario each device sends 180 messages in an hour to the IoT service. The data usage of 180 messages was 61.44 KB according to the Bluemix metering. This means that 0.3412 KB (i.e. 61.44 / 180) was logged by Bluemix for a message in contrast to the originally created text file with size of 0.053 KB. From this point, we can count that using 128 devices we have $128 * 24 * 180 * 31 = 17141760$ messages for the whole month. We can calculate the total data exchanged by multiplying the number of the total messages with the size of one message which results $17141760 * 0.3412 \text{ KB} = 5848768.512 \text{ KB} = 5711.688 \text{ MB}$. This is significantly larger than the estimated amount because of the additional information added to messages by Bluemix. Bluemix charges up 0.00097 Euro for each MB exchanged, so it means 5.54 Euro after that nearly 6 thousand MB exchanged. This price is also larger than the estimated one (~ 0.81 Euro) as well as the amount of data exchanged. The cloud side prices do not depend on anything as well as in the case of Scenario I. The conclusion is that we need to pay some more Euros than expected due to the larger message size.

This investigation suggests that we can save money by using an own gateway that is capable of aggregating sensor data, resulting in less message transfers. Instead of sending so many messages continually and generating a big amount of data traffic, we should use the own gateway to aggregate the data and send the collection of data to the cloud at certain times and not immediately.

6 CONCLUSIONS

IoT systems are very diversified and has a wide range of application fields ranging from wearable devices to wide-spread and distributed sensor networks. To implement these systems, it is essential to use a unified and secured data

management and processing solution that can be easily and efficiently supported by the federated solutions of compute and data clouds.

In this work, we defined two IoT Cloud scenarios to investigate cost estimations based on real-world examples that vary in the following parameters: total number of sent messages in a certain period of time, the number of devices. Our cost estimation results show that different providers apply different pricing schemes, and the actual costs very much dependent on the messages sent and data transferred. In general, we can say that Bluemix and Amazon is the cheapest due to the cheap message prices in case of Bluemix and the cheap virtual machine-related prices in case of Amazon. What is for sure, if we want to use a large number of devices, we should not choose Oracle because of its expensive device prices. Although, for a really small system like the first scenario, we can go with Azure.

We also implemented the second scenario at the one of the investigated providers, namely IBM Bluemix, to find out how the prices would work out actually, how the concrete bills would assimilate to our estimations. As a result, we can state that the estimations are basically similar to the actual prices measured. The actual prices slightly differ from the estimated ones and this difference comes from the larger message sizes during the transfer of the messages.

ACKNOWLEDGMENTS

The research leading to these results received funding from the European COST programme under Action identifier IC1304 (ACROSS).

REFERENCES

- [1] A. Botta, W. de Donato, V. Persico, A. Pescapé. On the Integration of Cloud Computing and Internet of Things. The 2nd International Conference on Future Internet of Things and Cloud (FiCloud-2014), August 2014.
- [2] S. Nastic, S. Sehic, D. Le, H. Truong, and S. Dustdar. Provisioning Software-defined IoT Cloud Systems. The 2nd International Conference on Future Internet of Things and Cloud (FiCloud-2014), August 2014.
- [3] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Comput. Netw.*, Vol. 54, No. 15, pp. 2787–2805, 2010.
- [4] Pricing of Microsoft Azure IoT Hub. Online: <https://azure.microsoft.com/en-gb/pricing/details/iot-hub/>, 2016.
- [5] IBM Bluemix IoT Platform. Online: <https://www.ibm.com/cloud-computing/bluemix/internet-of-things>, 2016.
- [6] Pricing of IBM Bluemix IoT Platform. Online: <https://console.ng.bluemix.net/?direct=classic/#/pricing/cloud-OEPaneId=pricing&paneId=pricingSheet>, 2016.
- [7] Pricing of Oracles IoT Platform. Online: https://cloud.oracle.com/en_US/opc/iot/pricing, 2016.
- [8] Pricing of Amazon IoT. Online: <https://aws.amazon.com/iot/pricing/>, Dec. 2016.
- [9] Pricing Calculator of Microsoft Azure. Online: <https://azure.microsoft.com/en-gb/pricing/calculator/>, 2016.
- [10] Pricing of Amazon EC2 Instance. Online: <https://aws.amazon.com/ec2/pricing/on-demand/>, 2016.
- [11] Pricing of Oracles Compute Service. Online: https://cloud.oracle.com/en_US/opc/compute/compute/pricing, 2016.
- [12] Pricing Calculator of Oracles Compute Service. Online: <https://shop.oracle.com/cloudstore/index.html?product=compute>, 2016.
- [13] Smart City of Messina. Online: <http://smartme.unime.it/>, 2016.
- [14] Official site of Idokep. Online: <http://www.idokep.hu>, 2016.