Multi-Cloud Management Strategies for Simulating IoT Applications

András Márkus, Attila Kertész

Abstract: Currently, the Internet of Things (IoT) paradigm is closely coupled with Cloud technologies, and the support for managing IoT data is one of the primary concerns of Cloud Computing. In IoT Cloud systems, sensors and different smart devices are connected to the cloud, and large amounts of data are generated by these things that need to be managed by infrastructure Cloud services. Simulation platforms have the advantage of enabling investigations of complex systems without the need of purchasing and installing physical resources. In our previous works, we chose the DISSECT-CF simulator to examine IoT Cloud systems, and developed cost-aware policies for managing IoT and Cloud components. The aim of this paper is to further extend the simulation capabilities of this tool by introducing multi-cloud management and selection approaches. We detail our proposed method for the extension, and evaluate multi-cloud utilization through a meteorological case study.

Keywords: Internet of Things, Cloud computing, Simulation

Introduction

In the paradigm of Internet of Things (IoT), sensors and smart devices are connected to the Internet giving way to many opportunities to use cloud and IoT services together [1]. Since more and more devices enter the network to form IoT systems, the dataflow and the workload of the supporting services are increasing. Hiring physical machines from virtual server parks fitting various IoT scenarios could be very expensive, and the investigation of IoT enabled cloud service compositions is not always possible with real cloud providers. As a result in many cases cloud simulators are applied to address such examinations with adequate results.

While network simulators could be too complex to simulate IoT and cloud systems together, due to detailed network configurations, special purpose cloud simulators may be over-tailored to cloud-specific details making it hard to express IoT needs. The number of IoT devices and usage areas are constantly growing, and some cases require immediate intervention after data processing, such as heart monitoring in smart homes, or traffic control in smart cities. This means we need new solutions and techniques for data storage, access and processing, which can be designed and evaluated in infrastructure cloud simulators extended with IoT simulation capabilities. Therefore we have chosen DISSECT-CF to perform our investigations [2].

In our earlier works we introduced and combined provider pricing schemes with IoT cloud management in DISSECT-CF [3]. Since cloud federations provide better services to users, the next step in our research is to enable the usage of multiple cloud datacenters to serve certain IoT scenarios. Therefore in this paper we introduce multi-cloud definition to DISSECT-CF, and propose three cloud selection strategies to be used for mapping IoT devices to cloud services. Finally, we evaluate our proposal through scenarios derived from a real-life weather forecasting service. The weather stations usually have different sensors and usage frequencies affecting data generating methods that can influence cloud service operation and also provider pricing.

Previous extensions of DISSECT-CF towards IoT

One of our main goals for choosing the DISSECT-CF cloud simulator for our investigations was its unified resource sharing mechanism. In this simulator we have two types of events: (i) recurrent time-dependent events that have a frequency value (e.g. 10 ms) which calls their methods regularly in every moment based on the given value, and (ii) the non-recurrent time-dependent events that have only a delay value (e.g. 5ms) denoting the time to be elapsed before its function has to be called. With these build-in events we can simulate the management of IoT systems including sensors and smart devices. The configuration of IoT system properties, such as network bandwidth, local repository size, operating time, target repository, number of sensors and frequencies can be done with an XML description. Our earlier extension also includes an application to simulate IoT data processing [3].

Considering provider pricing, another two XML descriptions can be used to set cloud side pricing and IoT side pricing. Usually the cloud side pricing is used to calculate the costs of virtual machines (VMs) used to run an IoT application. It defines a fixed monthly cost per VM instance, but some providers charge the hour per price for every instance the IoT application needs. To manage data coming from IoT devices and sensors, we need to calculate the IoT side costs, that can also be set based on real provider pricing schemes (e.g. Amazon, Azure, IBM and Oracle – as defined in [3]). In general, the IoT prices are calculated after the generated data traffic in MB following the "pay as you go" approach, while some providers charge after the number of messages exchanged in a month, or set a monthly device per price or messages sent in a day.

By executing a simulation, the following steps are taken: First, a cloud is set up using an XML description (we used the model of a Hungarian private infrastructure called the LPDS Cloud of MTA SZTAKI), and the necessary amount of stations are initialized, and the VM parameters are loaded from additional XML files, which also describe the cloud and IoT costs. Next, the application is started to deploy an initial VM and to start the metering and data generation processes of device stations, and to start IoT and cloud operation price estimation in parallel. During the application execution, a service checks if the cloud repository received a scenario-specific amount of data, if so, then a compute task will be generated which use the cloud resources for data processing. Finally, data generation by sensors, dataset allocation to virtual machines, compute task execution and possible starting and stopping of virtual machines are repeated till the end of the simulation.

The proposed cloud selection strategies

The main research question of this paper is how we can influence the behavior of an IoT application, if the sensors can have different allocation strategies for multiple clouds. In the earlier version of the extended DISSECT-CF we had only one cloud datacenter to start VMs, therefore all sensors and smart devices was connected to this specific cloud, and all the generated data of the sensors were processed by virtual machines running in the same cloud. This cloud had a preloaded cost calculation policy with a single pricing scheme. As a result, a single cloud could be easily overloaded, and the unprocessed data could hinder the operation of the IoT application causing longer response times, even service unavailability for real-time solutions.

In this work we introduce the possibility of the multi-cloud management for IoT cloud simulations in DISSECT-CF. During the start of the simulation we can set up different clouds using the extended XML description denoting sets of physical machines and repositories with various properties. We can associate different pricing policies to the defined clouds, and within a simulation the application can decide to which cloud the IoT devices should be connected, thus where the generated data should be sent and processed.

In the IoT paradigm the sensors are passive entities of the systems, thus their performance is limited by the operation frequency (i.e, data generating, storing, sending to the cloud), uptime and network connection. Usually large amounts of sensor data are sent from the smart devices to cloud resources for further computation and analysis. Since using these resources cost money, IoT system operators can reduce their expenses by selecting a provider having a suitable pricing scheme.

Within this research we defined three different strategies to perform cloud provider selection (to be done for each sensor start-up), which can be denoted by setting the *strategy* field of the XML description of each sensor. These strategies are the followings: (i) *random*: a sensor (or a set of sensors, i.e. a station) chooses one from all the available clouds randomly; (ii) the *cost-aware* strategy looks for the cheapest available cloud (based on their static pricing properties), thus it compares the prices of the required VM for a given sensor. Its algorithm first orders the cloud by their hourly usage prices, then by the VM instance prices. This solution may be more suitable for IoT applications having relatively small data processing needs. Finally, the last one is (iii) the *runtime-aware* strategy, where the corresponding algorithm ranks the available clouds by a specific value defined by the ratio of the number of already connected sensors and the number of the available physical machines of the given cloud. This is a dynamic strategy taking into account the actual load of the available clouds. Applications having longer data processing needs may prefer this strategy.

Evaluation with a weather forecasting scenario

One of the earliest examples of sensor networks comes from the field of weather prediction, therefore we chose to model the crowdsourced meteorological service of Hungary called *Idokep.hu*. It operates more than 400 stations generating sensor data (including temperature, humidity, barometric pressure, rainfall and wind properties), and the actual weather conditions are refreshed on its website in every 10 minutes.

In the scenario we use to model its operation, all stations have 8 sensors and the message size of the sensors can be set up to 0.05 KBs, and the sensors generate data in every minute. The start-up period of the stations were selected randomly between 0 and 20 min. In order to exemplify the usage of different cloud selection strategies, we defined periodic start-up and shut-down dates for certain stations (e.g. to represent malfunctions or failures). We simulate a whole day of operation (from 0:00 a.m. to 24:00 a.m.), and we start the simulation by setting up 200 stations (at 0:00 a.m.). At 2:00 a.m. we start 100 more, and at 10:00 a.m. 200 more to scale the total number of operated stations up to 500. At 2.00 p.m. we shut down 200 stations to scale down the number of running station to 300 by 10 p.m.

With these station management timings we run four different test cases: (i) all stations run with random strategy, (ii) all stations run with cost-aware strategy, then (iii) all stations run with runtime-aware strategy. Finally, (iv) we mixed the three techniques in the last experiment. For this evaluation we used three different clouds with different cloud-side pricing, but the same IoT-side pricing was set for handling the IoT devices in all test cases. Table 1 shows the detailed configurations.

We executed the formerly defined scenario with the four test cases. As we mentioned before, the IoT side cost is 0.352 Euro for all cases using the IBM Bluemix IoT provider. The results of the experiments can be seen in Figure 2. The so-called timeout parameter denotes how much time it took for the application to terminate (i.e. perform all remaining data processing) after the last station stopped working (at 24:00 a.m.).

Table 1: Detailed multi-cloud configuration for the evaluation

Cloud	Physical machine	Hourly price	VM type
LPDS-1	1 PM - 32 cores, 128 GB RAM 5 PMs - 8 cores, 12 GB RAM	0.297 Euro	8 core, 14 GB RAM Azure Large category
LPDS-2	1 PM - 48 cores, 128 GB RAM 5 PMs - 8 cores, 12 GB RAM	0.039 Euro	1 core, 2 GB RAM Amazon Small category
LPDS-3	1 PM - 64 cores, 128 GB RAM 2 PMs - 8 cores, 12 GB RAM	0.150 Euro	4 core, 2 GB RAM Bluemix Medium category

As we can see the cheapest solution for the current IoT application was the runtime-aware strategy with 41 Euros, utilizing 26 virtual machines. The cost of the cost-aware strategy is very close to the runtime-aware one, but it had a 40 minutes timeout, because of the lower speed provided by the weaker virtual machines. The presented scenario and the extended DISSECT-CF with the mentioned XML description formats are available at [4].

Conclusion

In this paper, we extended the DISSECT-CF simulator with the possibility of utilizing multiple clouds for IoT cloud experiments. We presented three different strategies for mapping IoT devices to cloud resources, and exemplified their utilization through a meteorological scenario. Our future work will address the development of other dynamic cloud selection algorithms using fuzzy methods.

Table 2: Evaluation results of the considered strategies

Strategy	**Random**	**Cost-aware**	**Runtime-aware**	**Mixed**
LPDS-1	26,135	0	21,977	23,166
LPDS-2	14,975	43,563	10,218	10,723
LPDS-3	21,750	0	9,140	14,549
Sum (Euros)	62,86	43,563	41,335	48,438
VMs	51	71	26	47
Timeout (min)	20	40	15	15

Acknowledgements

This work was supported by the Hungarian Government and the European Regional Development Fund under the grant number GINOP-2.3.2-15-2016-00037 ("Internet of Living Things").

References

[1] A. Botta, W. de Donato, V. Persico, and A. Pescape. Integration of Cloud computing and Internet of Things. Future Gener. Comput. Syst. 56, pp. 684-700, 2016.

[2] G. Kecskemeti. DISSECT-CF: A simulator to foster energy-aware scheduling in infrastructure clouds. *Simul. Model. Pract. Theory*, 58P2, 2015.

[3] A. Markus, A. Kertesz, G. Kecskemeti. Cost-aware iot extension of dissect-cf. *Future Internet*, 9(3), 2017.

[4] DISSECT-CF extensions towards IoT. Online: https://github.com/andrasmarkus/dissect-cf/tree/pricing. Accessed in March, 2018.