

Resource Optimisation in IoT Cloud Systems by Using Matchmaking and Self-management Principles

Martin Serrano¹, Danh Le-Phuoc¹, Maciej Zaremba¹, Alex Galis²,
Sami Bhiri¹, and Manfred Hauswirth¹

¹ National University of Ireland Galway, NUIG – Digital Enterprise Research Institute,
DERI, Galway, Ireland

{firstname.lastname}@deri.org

² University College London, UCL – Department of Electronic and Electrical Engineering,
London, U.K.

a.galis@ee.ucl.ac.uk

Abstract. IoT Cloud systems provide scalable capacity and dynamic behaviour control of virtual infrastructures for running applications, services and processes. Key aspects in this type of complex systems are the resource optimisation and the performance of dynamic management based on distributed user data metrics and/or IoT application data demands and/or resource utilisation metrics. In this paper we particularly focus on Cloud management perspective – integrating IoT Cloud service data management - based on annotated data of monitored Cloud performance and user profiles (matchmaking) and enabling management systems to use shared infrastructures and resources to enable efficient deployment of IoT services and applications. We illustrate a Cloud service management approach based on matchmaking operations and self-management principles which enable improved distribution and management of IoT services across different Cloud vendors and use the results from the analysis as mechanism to control applications and services deployment in Cloud systems. For our IoT Cloud data management solution we utilize performance metrics expressed with linked data in order to integrate monitored performance data and end user profile information (via linked data relations).

Keywords: Internet-of-Things, Service Platforms, Management, Linked Data, Cloud Computing Systems, Elasticity, Self-Management, Autonomic Management, Cloud Monitoring, Interoperability, Virtual Infrastructures.

1 Introduction

Cloud systems are offering powerful and flexible capabilities for running Internet of Things (IoT) data services and applications by using Internet infrastructure. This capability consists of facilitating service deployment on shared network and computing infrastructure resulting in reduced operational costs [1][2].

Cloud computing offers secure, reliable, scalable and elastic infrastructure operations. Differently from the conventional designs based on hosted and fixed servers solutions, Cloud computing allows a flexible world of options in configuration

and elasticity in eases expansion of Cloud computing resources as network, storage and computing requirements increase [1][3].

As one of the most prominent challenges in IoT Cloud systems, the control of their resources and the dynamic behaviour of virtual infrastructures rely on their capacity to run computer applications, services and processes independently. A key aspect in this type of complex systems is the performance of dynamic management based on distributed user data metrics, IoT application data demands and resource utilisation metrics (broadly used in managing communications networks). Efficiency of Cloud infrastructures and their autonomic management [4] are research challenges in today IoT Cloud systems [5]. Enabling elasticity of virtual infrastructures as a response to either load balancing protocols or remote monitored data processing is fundamental. In these settings managing Cloud services lifecycle by enabling scalable applications and using distributed information systems and linked data processing in a securely is crucial.

This paper introduces a novel approach for enabling elasticity of IoT Cloud services. The concepts and trends discussed in this paper can be applied to public, private and hybrid Clouds. This paper introduces a Cloud management perspective where integrated IoT Cloud service data management is based on annotated data of monitored Cloud performance and user profiles, enabling management systems to use shared infrastructures and resources to provide efficient deployment mechanism for IoT services and applications. Linked data mechanisms are used in this paper in the context of Cloud service management tools facilitating scalability, enabling efficient management and providing infrastructure control. An experimental testbed implementation using matchmaking requirements and distributed IoT data is illustrated in this paper. This paper briefly analyses and discusses technology trends in the area of Cloud computing and introduces linked data concepts and management principles to understand the advantages linked data offer for enabling matchmaking and services control on IoT Cloud service infrastructures.

This paper is organized as follow: In section 2 we introduce the matchmaking in the Cloud computing context. Section 3 discusses the state of the art related to IoT Cloud services and applications. Section 4 examines challenges and limitations on IoT Cloud computing. Section 5 describes Cloud service matchmaking evaluation and focus on IoT Cloud services. Section 6 describes scalable features about matchmaking and self-management and its benefits when used in IoT Cloud services. Section 7 presents a matchmaking scenario and IoT Cloud service control, from a point of view of on-demand scalability by using resource control by aggregated performance data and distributed computing monitoring. In Section 8 we conclude the paper and discuss future work.

2 Matchmaking for IoT Cloud Systems and Services

The Matchmaking process is an important component of cloud systems. One of its main purposes is to infer unknown preferences from annotated data of monitored performance and user profiles to another. In IoT effective realisation of IoT Cloud

service matchmaking will further improve adoption of Cloud computing solutions in this area and provide a consolidated view of the Cloud market enabling cost effective scalability by leveraging Clouds on-demand. IoT Cloud services need to be individually configured by allowing Cloud consumers to choose characteristics of provided Cloud offers. Service descriptions are the core artefacts used in Cloud service matchmaking and the matchmaking results depend largely on the provided service descriptions. IoT Cloud service matchmaking is a crucial part of the solution for self-healing Clouds [6] where alternative vendors are identified on-the-fly and used as replacements for original vendors which no longer satisfy consumer's requirements (e.g., incur higher cost or provide unreliable service). In order to realise this vision, service matchmaking has to operate on quantifiable artefacts suitable for an objective Cloud service ranking. *Service offers* [7] are the most specific, quantifiable, often request-dependent descriptions of Cloud service capabilities.

Details of Cloud services (e.g., the number of available cores, RAM, storage space, etc. in the case of IaaS resources) and service consumer requirements and preferences often change. There is *no one size fits all* Cloud vendor and the ability to determine the best (cheapest, most reliable) Cloud service offer provides a significant business value. Cloud computing vendors offer a great variability in their services. Their services are highly configurable and details of their offers (in the case of IaaS comprised of: availability, price, CPU, RAM, etc.) depend on user requirements. A service might offer an optimal solution for a certain requirements, while for different requirements it might be unavailable (e.g., due to the Cloud's physical location) or be uncompetitive (expensive, unreliable).

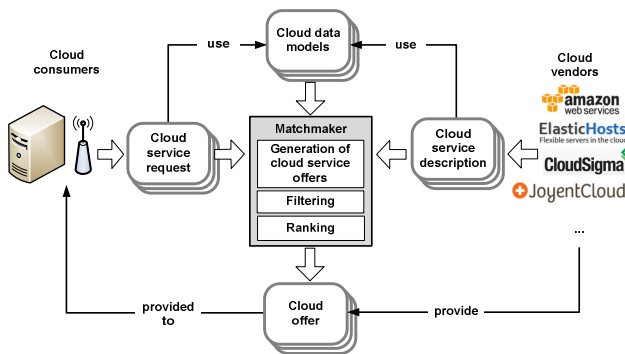


Fig. 1. Cloud computing matchmaking scenario

As shown in Figure 1 we distinguish three types of stakeholders in the IoT Cloud service matchmaking process: consumer (IoT user or vendor), matchmaker (broker) and Cloud vendors (IaaS public Clouds vendors). IaaS Cloud services are described in terms *service descriptions* and service consumers formulate their requirements in *service requests*. Both service descriptions and service requests are communicated to the matchmaker whose role is to provide the best match between the involved parties. Service offers are generated for a specific service request in the matchmaking phase and are often valid for a limited period of time only. The role of the matchmaker is to

facilitate generation of Cloud service offers, provide their filtering in order to remove offers that do not satisfy consumers' hard constraints and to finally rank them according to consumers' preferences.

3 Cloud Computing and IoT Services

The convergence of physical infrastructure network resources and computing mechanisms enable virtualization of all types of resources [3][8][9]. IoT as an emergent paradigm is based on the computing realization for heterogeneous infrastructure and the interoperability of diversity of data models. The main benefits of Cloud computing in IoT are: 1) facilitated on-demand service provisioning, 2) pay-as-you-go pricing models and 3) optimisation of shared resources, aspects that are described in Cloud computing systems and that everyday acquire more importance in the IoT domain.

In Cloud computing systems, the owner of the infrastructure is called Cloud vendor. Currently some of the main Cloud vendors are Amazon [5], Salesforce [10] and Google [11] leading the market with their large infrastructures and software portfolios. However increasing popularity of IoT paradigm leads to Cloud computing infrastructure implementations to be most reliable solution for IoT systems which subsidise services in the Cloud with the objective of reducing administration, maintenance and management cost.

IoT Cloud services benefits from Cloud computing by the easy administrative and technological on-demand expansion by running of shared infrastructure and by providing most of the time server applications according with user-oriented demands. Likewise IoT Cloud services benefits from resources on-demand processing, at the same time allowing users for managing processing sessions which, infrastructure can be dynamically assigned to other users or computing purposes, this feature turns on IoT Cloud services as an advantage on IoT Flexibility. Elasticity is an inherent Cloud computing feature that allows for efficient session activation and de-activation that is translated in expansion and reduction of utilized IoT Cloud infrastructure (physical or virtual). Thus elasticity enables scaling up and down IoT Cloud service according to the actual demand.

Cloud services are typically offered in a pay-as-you-go pricing model and are characterized by complex pricing models including time-based, utilization-based, and SLA-based charges. IoT Cloud services benefits of this on demand feature and rely on computing resources that are provided under an agreed Quality of Service (QoS) and Service Level Agreement (SLA). For example, Amazon charges for an instance based on its size and uptime, while allowing for a reduced payment if you pre-pay for a year or 3 years; inbound and outbound network traffic, with reduced pricing and increasing capacity [5]. Amazon also charges for the storage capacity, reliability level, transfer rate, & number of I/O operations performed. Network charges differ also on the basis of the physical location - being free in availability zone, reduced for inter-zone, and full across regions. Load balancing is charged in addition. Amazon also enables users to bid for EC2 spot instances representing computation resources, as such it uses the Cloud infrastructure to enable functionality and scalable services by using the Cloud infrastructure more efficiently.

Recently IoT companies interested in utilizing services in the Cloud have come to realize that Cloud computing can help them to expand their services efficiently and improve the overall performance of their current systems by building up an overlay support system. This provides increased service availability, task prioritization and service load distribution, all based on users' individual requirements and priorities.

4 IoT Data Cloud Challenges and Limitations

In this section we summarize a number of research challenges for IoT Data Cloud. It introduces important issues with respect to improving Cloud management for IoT applications. There are still number of issues preventing a wider adoption of the IoT Cloud computing that are summarized in this section. The Open Data Center Alliance¹ identified interoperability, service catalogues and standard units of measurement as the priorities that the Cloud community needs to resolve. IoT Cloud services are currently described informally, and determining and comparing on the same reference Cloud details (e.g., price per month, CPUs, RAM and storage capacities of the provided infrastructure, etc. in the case of IaaS). IoT Cloud is a time consuming and manual process what prevents Cloud consumers from gaining a consolidated view of the IoT Cloud data market. Figure 2 represent the most important challenges and limitations marked as hype research challenges in Cloud computing (i.e. Multitenancy, Elasticity, Transparency and Service Composition), same problems that IoT domain face up for enabling IoT data service management capabilities.

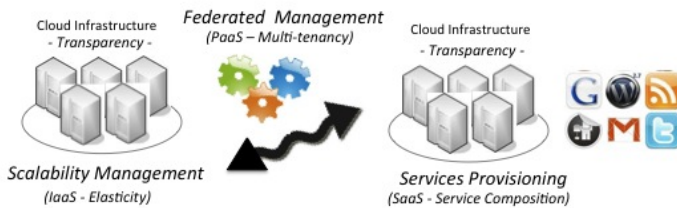


Fig. 2. Cloud Computing Challenges and Limitations

IoT Cloud data services and the use of Cloud infrastructures supporting IoT applications are becoming a common practice in industry (particularly in sensor-related data industry). This is the result of the Cloud economic advantages regarding the initial investment in infrastructure and the cost of outsourcing IoT Cloud data services [12]. Theoretically unlimited scalability of Cloud service provisioning and the simplicity it represents to users operating in the Cloud is attracting more and more businesses to switch to the IoT Cloud data infrastructures and services. The following are key management problems and aspects which would need innovative.

¹ <http://www.opendatacenteralliance.org>

- Real-time scalability of the IoT Cloud services. IoT Cloud services should reflect the current load and actively change the underlying infrastructure to address time-varying demand. Therefore the application is not static anymore and centred around a single IoT Cloud vendor but it can evolve over time across various vendors.
- Scalability of the IoT application. Adding more resources does not necessarily result in increased performance of the application. The challenge in this area is to find proper Cloud scalable application infrastructure as well as to perform predictions regarding application workload. However, Cloud services are commonly referred to as 'infinitely' scalable. That is not the necessary case for IoT, in some situations a collaborative solution of Cloud vendors might be the optimal Cloud implementation in terms of cost, scalability and key performance indicators.
- Optimisation for complex infrastructure and optimisation for complex pricing model, there are many possible options for application architecture. A challenge is to design an optimal architecture to fit Cloud environment and at the same time to optimally use all IoT resources with respect to pricing.
- Security, with the participation of multiple users into the IoT environment, the privacy and security of the information while running processes is crucial. Privacy on IoT Cloud data rely on strong encryption and protection protocols and are guarded by firewalls. However in other IoT security requires more research activities in terms of data protection algorithms and protocols for security and privacy of user identity.
- Multitenancy, as the number of partitioned virtualized resources (or instances) grows the challenge of how to optimally allocate them to multiple tenants become more apparent. This can be addressed by allowing users to manage their own delegated resources – non-expert resource management. This raises a number of challenges including: supporting non-expert resource monitoring and management, managed delegation of management resources and authorities, and auditing.
- IoT Cloud Management, Cloud computing takes advantage of the computing parallelism where by definition it is an attractive feature to optimize the usage of the computing resources and for this reason recently catalogued as green ICT. For IoT Cloud data, this optimisation and management functionality is translated into economic benefit. For this and other economic reasons physical infrastructure has migrated to the Cloud and likewise IoT Cloud data.

The challenges for IoT Cloud data systems are without doubt that must be able to support Clouds system up to their breaking point. In this respect, for example in Cloud elasticity there are two main metrics to be considered: spin-up elasticity (the time response between the $t=0$ when computing resources request and $t+n$ the time when resource request is running) and spin-down elasticity, (the time response between $t=n$ no longer requiring compute power and no longer paying for it) [13][14].

5 IoT Cloud Service Matchmaking Evaluation

In context of IoT Cloud computing research, we have applied our Service OFFer Discovery (SOFFD) [7] approach to the IaaS Cloud computing domain, made it available online² and evaluated it. Service OFFer Discovery (SOFFD) is a generic service search engine for discovering, configuring and selecting highly configurable, search request-dependent services having dynamic properties. It operates on lightweight service description and search request models grounded in RDF and SPARQL which refer to Linked Data vocabularies and datasets in order to foster service description reuse and to facilitate service matchmaking. SOFFD generates request-dependent service offer descriptions tailored to the search request at hand and provides matchmaking against these dynamically generated service offers.

We have described with Linked Data and integrated Cloud services from nineteen IaaS vendors. It has been evaluated by running a series of tests with different search requests and IaaS services. Each request varied by a different in its complexity, specified hard constraints and preferences.

Flexible Cloud search requests (e.g., CPU between 2-4 ECU) enable generation of large number of service offers and required for the optimal search for the best service offers among them. Search requests with fixed properties (e.g., CPU has to be 4 ECU) yield far less service offers, whereas SOFFD allows for the generation of a large number of service offers what plays an important role in finding optimal service offers. For example, fixed search request properties may not yield any results at all whereas flexible search requests are more likely to yield promising results.

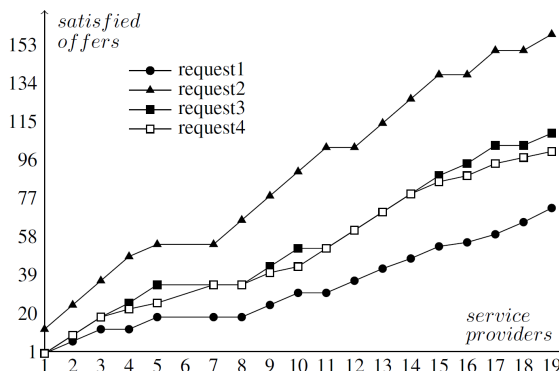


Fig. 3. Generated IaaS Cloud Computing Offers

We show a number of generated service offers for our search requests in Figure 3. As we can see in this Figure, a single search request results in a large number of generated Cloud service offers. SOFFD dynamically generates, filters and ranks a number of Cloud service offers. Cloud service offered by a single vendor

² <http://soffd.derii.ie/>

(e.g., Amazon) when evaluated against flexible search request results in a number of possible service offers that SOFFD matchmakes.

SOFFD discovery algorithm scales linearly with a number of generated offers. Evaluation of SPARQL-based hard constraints and calculating ranking takes most of the computational time. We provide IaaS matchmaking results to the service consumer asynchronously what limits perceived time overheads. Our scalability results could be further improved by utilising caching of service offers. Currently SOFFD always generates all possible Cloud service offers for each individual search request.

6 Linked Data and Scalable Cloud Management

In this section we analyse features of Cloud management and infrastructures pointing towards the advantages that linked data can offer when used with the Cloud-based Service management control loop. In Cloud management, in order to apply deployment policies for a service request, the Cloud manager has to check the logs of running services. Traditionally, the logs recorded from running services and computing nodes are processed in the Data Correlation Engine under separated files that need to be loaded to relational tables to be accessed by a rule engine. However, the logs and service requests are fed into Data Correlation Engine in the stream fashion. There are a wide variety of literatures in stream/event processing studied in [15][16] showing that relational database engines do not perform well with continuous processing over log streams. Therefore, the stream/event processing engine should be integrated into the Data Correlation Engine.

In addition, to support the seamless integration of rule-based policy represented as Semantic Web Rule language (SWRL) on top of background knowledge represented in Description Logic with log streams, we propose using linked data model for the Data Correlation Engine. In this unified model, all the data is represented as a graph within the layered system representation as it is illustrated in Figure 4. This graph is composed from RDF triples which can be used to represent data, knowledge base and rules. The upper layer is for static data such as ontologies, schema and business logics which is time-independent. The lower layer is used for linking data of log streams into vocabularies and logic rules in the upper layer. The linked data in the lower layer is called linked data stream as depicted in Figure 4.

The linked stream data model brings several advantages in data correlation operations. The first advantage comes from the data distribution. The graph-based layout gives the data processing operators the global view of the whole dataset. Therefore, the query processor can filter the irrelevant data to a query much earlier than the log-file approach does. Traditionally, the monitoring data recorded in separated log files are partitioned in individual services, processes, etc, thus, cross-correlating the relevant data items among them needs to load all the data into a relational storage before carrying out the correlation.

The push-based and incremental processing model of linked stream processing engines provides much better performance than that of traditional relational database

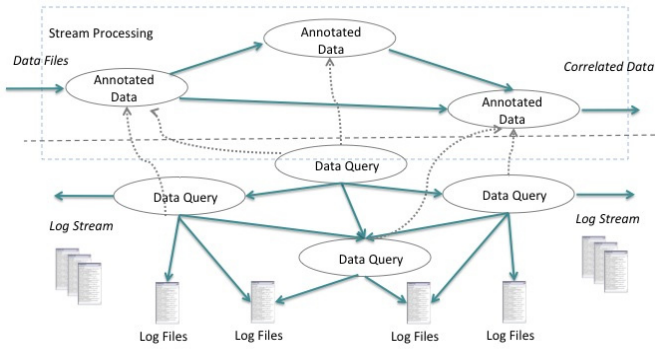
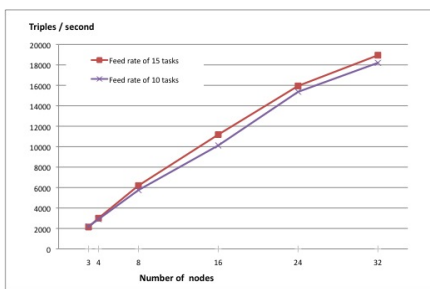


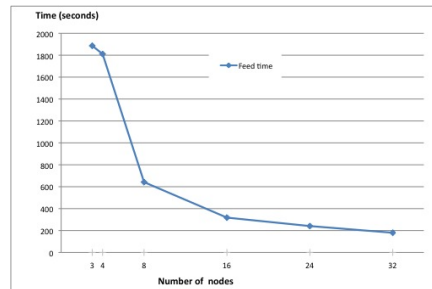
Fig. 4. Cloud Management Service Control Loop

engine. Because a query over the log streams on relation database is performed in pull-based and one-shot fashion whereby any new snapshot of log stream needs the full computation. Thanks to the push-based and RDF triple data model, the log data can be pushed gradually per triples or a set of triples into the Data Correlation Engine. This helps to avoid the overload of matching schema and data loading when receiving large IoT Cloud computing monitoring logs.

To meet the query processing demand of Data Correlation Engine, we have developed and evaluated our Continuous Query Evaluation over Linked Stream (CQELS) engine [16]. This engine can consume very high throughput from log streams and can access large persistent triple storages with millions of triples. The current version can deal with thousands of concurrent queries corresponding to service matching policies registered. The Figure 5 shows initial experiments with stream data processing. Figure 5(a) represent “feed rate” defined as throughput of data consumption (from raw to triples) (10/15 tasks are experiments that have 10/15 concurrent feeding tasks on one node). Figure 5(b) shows the streaming data response with time reference and increase in number of nodes or “feed time” defined as the time spent for loading and transforming a batch of raw data. In summary, more nodes give more capacity to process longer streaming data files and similarly more nodes less result in less time required to satisfy query requests on streaming data.



(a)



(b)

Fig. 5. Stream Data Processing Analysis

7 Matchmaking and IoT Cloud Service Control

Matchmaking of Cloud services and resources is addressed in [17] where authors propose a model for reflecting dynamic information (e.g., changing Cloud capabilities, response times, etc.) in Web service descriptions and updating WSDL with dynamic resource attribute information. SOFFD search request model provides more advanced support for search criteria by supporting dynamic generation of service offers and handling both hard constraints and preferences.

Matchmakers utilizing Linked Data standards can be easily provided for third-party integration and used in automated processes (e.g., for hybrid Cloud dynamic scaling using the most preferable Cloud vendor). SOFFD as an example of semantic service matchmaker applied to the Cloud computing area allows consumers to dynamically determine the best available service offers and provide a consolidated view of IaaS market. Matchmaking cannot be achieved using high level service descriptions, as Cloud service properties (e.g., price, provided CPU, RAM, storage, etc. in the case of IaaS) are either offered in bundles or are configurable. Prices of Cloud services are request-dependent and can be dynamic (e.g., price of Amazon EC2 spot instances fluctuate depending on supply and demand). From the point of view of Cloud consumers, matchmaking of Cloud service offers is far more beneficial than operating on the level of high-level Cloud descriptions. Only service offers can satisfy the concrete needs of Cloud consumers.

In Cloud computing, highly distributed and dynamic elastic infrastructures [18] are deployed in a distributed manner to support service applications. In consequence development of management and configuration systems over Internet automatically controlling virtual infrastructures are necessities [4]. Management operations modifying the service lifecycle control loop and satisfying user demands about quality of service and reliability play a critical role in this complex management process. Cloud computing typically is characterized by large enterprise systems containing multiple virtual distributed software components that communicate across different networks and satisfying secure and personalized service requests [14][19]. The complex nature of these user requests results in numerous data flows within the service components and the Cloud infrastructures that cannot be readily correlated with each other. Ideally data flows can be treated at runtime by correlation engines, given thus the possibility of multiple Cloud infrastructures and beyond boundaries the free data exchange and also cooperate to serve Cloud common services.

A Cloud management service control loop is depicted in Figure 6. From a data model perspective, this control loop on-demand scalability and scalability prediction addresses computing data correlation between performance data models of individual components and service management operations. Exact component's performance modelling is very difficult to achieve since it depends on the multitude of variables. To simplify this complexity we focused the model on performance values such as available memory, CPU usage, system bus speed, and memory cache thresholds. Instead of exact performance model we use an estimated model calculated based on monitored data from the Data Correlation Engine represented in the Figure 6 [20].

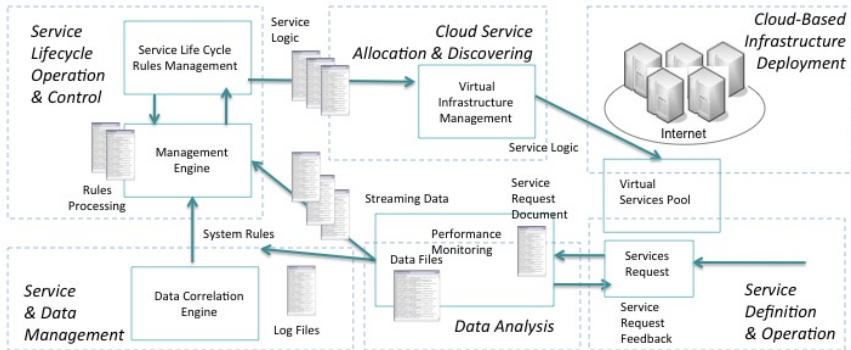


Fig. 6. Cloud Service Management Control Loop

An intermediate solution to efficient service control management is used, the data logs used by the Service Lifecycle Rules Manager is compared and linked with data semantic engineering mechanisms to enable service lifecycle control operations. Result of this linked data operation is named event data model, which is a standardized model and can be used indistinctly in service management or network infrastructure management domains.

These standard data models can be understood by diverse infrastructure and platforms modifying or adapting their performance according to particular applications and systems and pre-defined operation requirements. For example, Management Policies Engine can use the Event data model to define changes and dynamically perform Cloud infrastructure control. Linked data standard model facilitates the dynamic adaptability of the information over infrastructures where other data models are incompatible. The main objective for using this standard linked data model is to modify the performance of the infrastructure satisfying thus general business goals according to the high level rules known as goals and defined into service contracts or service agreements. A service translation is needed in terms of defining the Service Logic that a Virtual Infrastructure Manager can understand.

The data link process provides assurances for interoperability between performance information and the simplistic commands contained in the data models. Further experimentation is being conducted to define service-discovering mechanism by using service allocation protocols.

The approach of using data model translation of the service allocation data model and perform linked data is being considerate as a tentative approach to this complex problem. The Figure 7 shows performance metrics in managing system with virtual infrastructures. Figure 7(a) represents initial experiments with semantic rule-based engine 2SCE [21], CPU usage percentage vs. the number of operations and service rules that has been created and processed by the 2SCE control engine. Figure 7(b) shows memory usage in megabytes vs. the number of operations and service rules when an application or service is using the loaded rules.

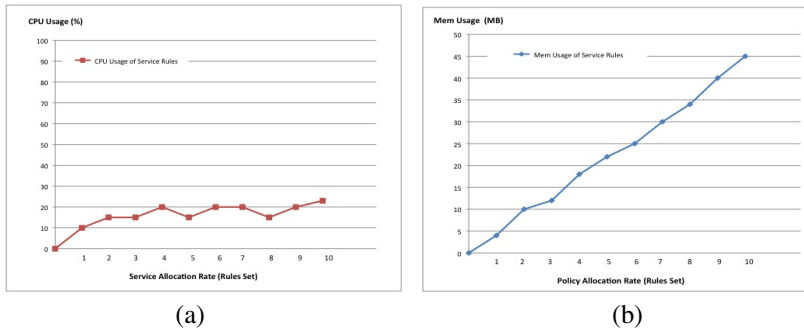


Fig. 7. Semantic-Based Rule Engine Performance Analysis

8 Conclusions and Future Work

In this paper we analysed key developments in IoT Cloud data including concepts, trends, challenges and limitations in this emerging area. One of the most important research challenges in IoT Cloud applications and Cloud data systems is the dynamic control of elasticity of the Cloud infrastructure utilizing performance metrics (i.e. logs from computing applications) and user requirements (i.e. annotated data as streaming data analysis).

We have introduced and discussed advancements in terms of using matchmaking and self-management principles in Cloud data added by link data mechanisms and related to IoT systems. Implemented architectural components, in the form of SOFFD prototyped middleware solution have been presented and evaluated as part of proof of the concept process enabling Cloud infrastructure selection. In addition introducing linked data mechanisms allow maximising control of service lifecycle. The concepts presented in this paper can be applied to IoT Cloud applications or Cloud systems.

Further work includes experimentation with service-discovering mechanisms by using service allocation protocols and embedded optimisation methods and algorithms for link data usage in IoT Cloud service management.

Acknowledgements. This work has been partially supported by the European Commission FP7-ICT-2011-7-287305 OpenIoT project³ and the Universal FP7 project⁴. It is also partially supported by Science Foundation Ireland under grant number SFI/08/CE/I1380 (Lion-II) CSET-DERI Centre for Science, Engineering and Technology - Digital Enterprise Research Institute.

Open Access. This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

³ OpenIoT Project <http://www.openiot.eu>

⁴ Univerself Project <http://www.univerself-project.eu/>

References

- [1] The Economics of the Cloud,
<http://www.microsoft.com/presspass/presskits/Cloud/docs/>
(online access Wednesday, January 05, 2011)
- [2] The 'InterCloud' and the Future of Computing, an interview: Vint Cerf at FORA.tv, the Churchill Club, SRI International Building, Menlo Park, CA (January 7, 2010),
<http://www.youtube.com/user/ForaTv#p/search/1/r2G94TmcUuY>
(January 2011)
- [3] Rochwerger, B., Caceres, J., Montero, R.S., Breitgand, D., Elmroth, E., Galis, A., Levy, E., Llorente, I.M., Nagin, K., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F.: The RESERVOIR Model and Architecture for Open Federated Cloud Computing. *IBM Journal of Research and Development* 53(4) (2009)
- [4] Serrano, J.M.: *Applied Ontology Engineering in Cloud Services, Networks and Management Systems*, 222 pages. Springer Publishers, Hardcover (2012) (to be released on March 2012), ISBN-10:1461422353, ISBN-13:978-1461422358
- [5] Amazon Web Services, <http://aws.amazon.com/>
- [6] Dai, Y., Xiang, Y., Zhang, G.: Self-healing and Hybrid Diagnosis in Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) *Cloud Computing*. LNCS, vol. 5931, pp. 45–56. Springer, Heidelberg (2009)
- [7] Zaremba, M., Vitvar, T., Bhiri, S., Hauswirth, M.: Service Offer Discovery Using Genetic Algorithms. In: *IEEE European Conference on Web Services, ECOWS* (2011)
- [8] Chapman, C., Emmerich, E., Marquez, F.G., Clayman, S., Galis, A.: Software Architecture Definition for On-demand Cloud Provisioning. *Springer Journal on Cluster Computing* (May 2011), doi:10.1007/s10586-011-0152-0
- [9] Clayman, S., Galis, A., Mamas, L.: Monitoring Virtual Networks. In: *12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010) - International on Management of the Future Internet*, Osaka, April 19-23, pp. 19–23 (2010),
<http://www.man.org/2010/>
- [10] Salesforce.com, <http://www.salesforce.com/Cloudcomputing/>
- [11] Google App Engine, <http://code.google.com/appengine/>
- [12] Adoption of Cloud Computing, in *Technology, Media & Telecom* by askvisory,
<http://askvisory.com/research/adoption-of-Cloud-computing/>
(online access Thursday, February 10, 2011)
- [13] Clayman, S., Galis, A., Toffetti, G., Vaquero, L.M., Rochwerger, B., Massonet, P.: Future Internet Monitoring Platform for Computing Clouds. In: Di Nitto, E., Yahyapour, R. (eds.) *ServiceWave 2010*. LNCS, vol. 6481, pp. 215–217. Springer, Heidelberg (2010)
- [14] Shao, J., Wei, H., Wang, Q., Mei, H.: A runtime model based monitoring approach for Cloud. In: *IEEE 3rd International Conference on CLOUD 2010*, pp. 313–320 (July 2010)
- [15] Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) *ISWC 2011, Part I*. LNCS, vol. 7031, pp. 370–388. Springer, Heidelberg (2011)
- [16] Le-Phuoc, D., Quoc, H.N.M., Parreira, J.X., Hauswirth, M.: The Linked Sensor Middleware: Connecting the real world and the Semantic Web. In: *9th Semantic Web Challenge co-located with ISWC 2011*, Bonn, Germany, October 23-27 (2011)
- [17] Goscinski, A., Brock, M.: Toward dynamic and attribute based publication, discovery and selection for Cloud computing. *Future Generation Comp. Syst.* 26(7) (2010)

- [18] Chapman, C., Emmerich, W., Galn, F., Clayman, S., Galis, A.: Elastic Service Management in Computational Clouds. In: 12th IEEE/IFIP NOMS2010 / International Workshop on Cloud Management (CloudMan 2010), Osaka, April 19-23 (2010)
- [19] The Real Meaning of Cloud Security Revealed (online access Monday, May 04, 2009), <http://devcentral.f5.com/weblogs/macvittie/archive/2009/05/04/the-real-meaning-of-Cloud-security-revealed.aspx>
- [20] Holub, V., Parsons, T., O'Sullivan, P., Murphy, J.: Run-time correlation engine for system monitoring and testing. In: ICAC-INDST 2009: Proceedings of the 6th International Conference Industry Session on Autonomic Computing, pp. 9–18. ACM, New York (2009)
- [21] Keeney, J., Conlan, O., Holub, V., Wang, M., Chapel, L., Serrano, M.: A Semantic Monitoring and Management Framework for End-to-end Services. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Management – IM 2011, Dublin, IE, May 23-27 (2011)