A Monitoring and Analysis Framework to Support Self-management in Cloud Application Platforms

Rustem Dautov[1, 2], Iraklis Paraskakis[2], Mike Stannett[1]
[1] *Department of Computer Science, Faculty of Engineering, The University of Sheffield.*
[2] *South-East European Research Centre, International Faculty of the University of Sheffield, CITY College, Thessaloniki, Greece.*

Abstract

Cloud computing in general and its Platform-as-a-Service segment in particular are becoming the *de facto* way of running enterprise-level software. By providing hardware infrastructure and support for software development, cloud application platforms exempt companies from major up-front investments, thus saving time and money. Additionally, they offer developers a range of generic reusable services, ready to be integrated into users' applications. However, to support such a flexible software development model, cloud providers have to exercise constant control over all critical activities taking place on the platform so as to prevent millions of deployed applications coupled with hundreds of add-on services from quickly dissolving into tangled and unreliable environments. To address this challenge, our approach combines techniques from the domains of Semantic Sensor Web, Stream Processing and Big Data analytics to create a monitoring and analysis framework and support autonomy in cloud application platforms. The approach relies on annotating monitored heterogeneous values with uniform semantic descriptions and applying run-time formal reasoning to these data streams so as to detect and diagnose critical situations.

Keywords Cloud Application Platform; Framework; Monitoring; Semantic Sensor Web; Sensor Analysis.

1. INTRODUCTION

With Platform as a Service (PaaS), subscribers pay for cloud resources when they are really needed, without the need to own or manage the underlying hardware infrastructure. Following the principles of Service-Oriented Computing, cloud application platforms (CAPs) also offer their subscribers a selection of pre-existing and reusable services, which can be seamlessly integrated into users' applications. This allows developers to concentrate on their immediate business tasks and exempts them from 're-inventing the wheel'. Such basic services, for example, may include data storage, queue messaging, searching, logging and caching, etc. However, as CAPs are becoming increasingly complex, dynamic and heterogeneous, one major challenge faced by CAP providers is the need for appropriate mechanisms for run-time monitoring and analysis to support timely detection and diagnosis of potentially critical situations. The goal is to enable self-managing behaviour in CAPs and prevent them from quickly dissolving into tangled unreliable environments [2].

To date, attempts to equip clouds with autonomic behaviour have mainly focused on the infrastructure level, by developing mechanisms for load balancing and elasticity [1]. Self-management at the PaaS level is still immature, and existing approaches tend to suffer from rigid, hard-coded architectures. Accordingly, our research effort aims at developing a monitoring and analysis framework to support autonomic capabilities of CAPs by following a declarative approach to the definition of knowledge and utilising existing techniques from the research areas of Semantic Sensor Web, Stream Processing and Big Data analytics. The main idea of this novel approach is to encode monitored heterogeneous data using Semantic Web languages. This will aid us to integrate these semantically enriched observation streams with static ontological knowledge and to apply intelligent reasoning.

2. OUR APPROACH

Our monitoring and analysis framework implements the established MAPE-K (Monitor, Analyse, Plan, Execute, and Knowledge) reference model for self-adaptations. A fundamental underpinning principle of our approach is the interpretation of CAPs as distributed networks of 'software sensors' – that is, services, deployed applications, CAP components, *etc*., which continually emit raw heterogeneous data to be monitored and analysed to support run-time situation assessment. We leveraged on this to apply existing solutions from the Semantic Sensor Web (SSW) community, which combines ideas from two research areas, the Semantic Web and the Sensor Web. This novel combination facilitates situation awareness by providing enhanced meaning for sensor observations. We were inspired by the Semantic Sensor Network (SSN) approach to express heterogeneous sensor values in terms of Resource Description Framework (RDF) triples using a common ontological vocabulary, and have created our own Cloud Sensor Ontology (CSO) to act as the core element of the analysis and monitoring framework for CAPs [3]. The CSO is used to support both self- and context-awareness of managed elements by describing the adaptation-relevant aspects of the managed cloud environment. The CSO also serves as a common vocabulary of terms, shared across the whole managed system and used as 'building blocks' in the three components of the framework (as depicted in Fig. 1):

1. **Triplication engine:** this consumes and 'homogenises' the incoming stream of raw heterogeneous monitored data

by representing data uniformly in the form of RDF triples constructed from the concepts defined in the CSO. The use of data streams consisting of time-stamped triples, incorporating OWL-based subjects, predicates and objects, promotes human-readability. It also allows exploitation of the extensive capabilities of SPARQL query languages.

2. **Continuous SPARQL query engine:** this supports situation assessment by reading the continuous RDF data streams and evaluating them against pre-registered continuous SPARQL queries. Thus, we are able to detect complex event patterns representing critical situations with minimal delay. Employing existing RDF streaming engines with support for 'on-the-fly' analysis of constantly flowing observations from hundreds of sensors is expected to help us achieve near real-time behaviour of the framework.

3. **Semantic Web Rule Language (SWRL) reasoning engine:** this generates a final diagnosis and an appropriate adaptation plan for an observed critical situation by applying formal reasoning over a set of pre-defined set of SWRL diagnosis policies. Thanks to the built-in reasoning capabilities of OWL and SWRL, the routine of reasoning over a set of diagnosis and adaptation alternatives is done using an existing, tested, and optimised mechanism. This also enhances opportunities for reuse, automation and reliability.

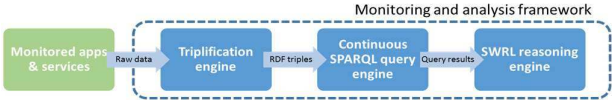

Figure 1. Three main information processing steps within the monitoring and analysis framework.

3. Proof of Concept and Evaluation

To demonstrate the viability of the proposed approach with a use case, we developed a prototype version of the framework [4], and provided it with raw sensor values coming from a Heroku[1] application, coupled with data storage, caching, worker and e-mail notification add-on services. Each of these services has associated resource quotas, which correspond to a selected subscription plan and can be seen as Service Level Agreements (*e.g.*, storage service users are provided with a certain amount of disk space, measured in MBs). Accordingly, the goal of the use case was to constantly observe the utilisation of the services and detect potentially critical situations whenever resource consumption was nearing its maximum allowed value (so as to prevent the system from crashing). Initial experiments demonstrated that the framework was able to collect and interpret monitored data from the application and also detect critical situations in a timely manner – *i.e.* within 1 second with up to 100 triples generated per second.

However, the performance, limited by hardware resource constraints, dropped with the increasing rate of incoming RDF triples. This can be rationalised by the general scalability issue of the formal reasoning processes, as well as the immature state of the RDF stream processing research area. To cope with this challenge, we applied the two general principles of Big Data processing – data fragmentation and parallel processing. We split the RDF stream into four sub-streams (each corresponding to a separate add-on service), which were then processed separately by a dedicated instance of the framework. To implement this distributed architecture, we deployed our experiment on IBM Streams[2], a software platform for the development and execution of applications that processes large amounts of continuously flowing data in a distributed and parallelised manner. Benchmarking and comparing the parallelised and the initial single-stream deployments demonstrated that detection time decreased by approximately 50% (with over 1000 triples generated per second).

4. Conclusions

By semantically annotating heterogeneously monitored values, the framework combines RDF triple streams with static ontological knowledge and performs run-time formal reasoning. Furthermore, it is based on an extensible, modular, and declarative architecture, which enables dynamic analysis and problem diagnosis to suggest further adaptation actions. Another benefit of the presented approach is its potential to address the scalability issue associated with in-memory processing of large amounts of streaming data – a challenge widely known and addressed by the Big Data research community. With data stream fragmentation and parallel processing, the framework is able to demonstrate stable results and cope with thousands of triples collected per second.

References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. Above the Clouds: A Berkeley View of Cloud Computing. 2009.
2. Dautov R, Paraskakis I, Stannett M. Utilising stream reasoning techniques to underpin an autonomous framework for cloud application platforms. J Cloud Comput; 2014.
3. Dautov R, Paraskakis I, Stannett M. Cloud Sensor Ontology and Linked Data to Support Autonomicity in Cloud Application Platforms. In: Klinov P, Mouromtsev D, editors. Knowl. Eng. Semantic Web, Springer; 2014.
4. Dautov R, Paraskakis I, Kourtesis D, Stannett M. Addressing Self-Management in Cloud Platforms: a Semantic Sensor Web Approach. Proc. Int. Workshop Hot Top. Cloud Serv. HotTopiCS 2013, Prague, Czech Republic: 2013.

[1] http://www.heroku.com

[2] http://www.ibm.com/software/products/en/infosphere-streams