Provenance-Aware Self-Healing Systems for Heterogeneous Computing Environments

Bahadır Dündar
Software Testing and Quality Evaluation Laboratory
TUBITAK
Gebze-Kocaeli, Turkey
email: bahadir.dundar@tubitak.gov.tr

Mehmet S. Aktas
Computer Engineering Department
Yildiz Technical University
Istanbul, Turkey
email: mehmet@ce.yildiz.edu.tr

***Abstract*— Dependability is an important attribute for heterogeneous computing environments and their applications. The growing complexity and dependency of heterogeneous computing environments makes fault tolerance an appealing research area. In this study, we discuss the inability to forecast faults in large-scale execution traces. In addition, we discuss research challenges in self-healing capabilities for autonomic, dynamically coordinated smart-environments based on the supervision of continuous monitoring of execution traces. To address such limitations and research challenges, we introduce a methodology, in which the state data coming from heterogeneous computing environments, such as Internet of Things (IoT) devices, is monitored for predictive maintenance, optimization and dynamic provisioning.**

Keywords-self-healing capabilities; fault tolerance; dynamic replication; provenance; heterogeneous; IoT

I. INTRODUCTION

IoT depends on self-configured smart objects that have limited storage and processing capacity. These small objects are dynamically coordinated in a large-scale environment [1]. Platforms for connected smart objects are built by plugging heterogeneous computational entities together in highly dynamic configurations. Orchestration, management and monitoring of such devices and smart objects are fundamental fields of research, as the number of interconnected objects is supposed to reach several hundred billion. This brings up the need for suitable approaches to adaptation, reconfiguration and self-healing systems, made of entities whose common characteristic is precisely their heterogeneity. The current state of the art in these applications lacks self-healing capability, which is commonly used to refer the capability of self-recovery of systems. To achieve this capability, there are number of coordinating nodes to perform a particular task, running on heterogeneously distributed computing platforms whenever an adaptation is required to an abnormal situation.

In this paper, our first goal is to investigate research opportunities in self-healing capabilities of dynamically coordinated heterogeneous distributed computing environments based on the supervision of continuous monitoring of execution traces. To this end, we use provenance as the descriptor metadata of the execution traces taken from IoT application nodes. Our next goal is to propose a software architecture for fault forecasting/estimation on large-scale execution trace data. In order to address these goals, this paper identifies following concrete research objectives described as follows.

Objective 1: To determine how to achieve fault tolerance to support self-healing capabilities in heterogeneous computing environments.

Objective 2: To determine how to enable fault forecasting/estimation within the execution traces of activities happening among IoT application nodes.

Objective 3: To determine how to optimize self-healing capabilities by taking into account both user involvement and computing environment in heterogeneous distributed computing environments [2].

This paper introduces architectural guidelines for providing fault tolerance to heterogeneous computing environments, such as IoT application domains. To achieve fault tolerance, the use of provenance metadata is proposed.

The rest of the paper is organized as follows. Section II presents the literature summary. Section III presents various application scenarios to describe the scope of this research. Section IV presents our proposed system architecture for developing fault tolerance in an IoT application domain. Finally, Section V presents conclusion and future work of our paper.

II. LITERATURE SUMMARY

In a typical IoT application, a smart object is a lightweight component that has a clear, software-defined API through, which it can be controlled and managed at runtime, and dynamically provisioned in an elastic way. Autonomous composition of these smart objects leads to complex software ecosystems. In autonomous heterogeneous computing environments, such as IoTs, there are different units that can potentially be provisioned at runtime. Currently, there is a lack of adequate solutions to achieve resilient, dynamically coordinated IoTs.

The IoT components of these applications have end-to-end links and data storage with read/write access. We argue that in the IoT domain, if a number of IoT devices or IoT services has faults, these faults will lead to complete failure of the entire IoT application. Since our study primarily focuses on fault tolerance mechanisms for heretogenous computing environments, such as the IoT, we only review background work on fault tolerance for these applications

 ISBN: 978-1-61208-474-9

running in heretogenous computing environments and consisting of different kinds of resource-limited devices. There are a number of previous studies that emphasize the importance of self-healing capability in IoT domains [3][4]. In light of this emphasis, we categorize and review the previous work as in the following paragraphs.

Deployment of IoT devices can be challenging. Fault tolerance has been addressed in several studies in this domain. These studies require deployment and re-configuration of the devices during the execution of IoT applications. However, these deployments require human intervention and must be performed by experts. In our study, we are interested in providing fault tolerance mechanisms that can run applications continuously, even in the case of individual node failure. Our approach is designed to run applications without stalling them. In this scenario, an IoT application can degrade gracefully under individual faults, but it can continue its execution.

In order to provide fault tolerance in the IoT domain, previous studies have used data replication techniques [5][6][7]. These studies have utilized both predefined replication and dynamic replication techniques. However, apart from the previous work, in our study we only focus on providing fault tolerance for services (instead of data replication) that are taking place in IoT applications.

Another approach for fault tolerance focused on service replication technique [8]. This was addressed for failover purposes. This approach only takes user requirements into consideration in deploying services onto multiple devices in order to recover failed services. In addition, this mechanism is tightly coupled with a middleware, and the number of replicated services is predefined. This approach does not support dynamic replication of services. In our solution, we introduce a loosely coupled fault tolerance mechanism to solve this problem. Our study aims at using a combination of both permanent and dynamic replication methods in order to optimize fault tolerance strategy in IoT domains.

With the increasing number of security attacks in the IoT domain, developing detection and prevention systems to protect the components has become essential [9]. There are some studies on detecting security attacks in the context of IoT [10][11][12]. We, however, are interested in the continuity of the entire IoT application, even under the condition of failure of individual work items. We are not concerned with preventing failures that may happen in individual IoT devices due to security attacks**.**

Arjun et al. proposed a framework for IoT devices in which these IoT devices can manage themselves with regard to their configuration and resource utilization [13]. However, this study focuses on a self-managing mechanism for individual IoT devices by controlling their behaviors. Additionally, this mechanism does not provide fault tolerance for entire IoT applications. Our study primarily focuses on fault tolerance for IoT applications, including multiple devices, which are coordinating with each other.

Self-healing systems should have the ability to protect themselves from possible failures. One of the methods of protecting systems from failures is to predict faults before they occur. There various types of fault prediction modeling techniques, such as Linear Regression, Naive Bayes Logistic Regression, Random Forests, Support Vector Machine and C4.5 are used in fault prediction [14][15][16]. These modeling techniques use different metrics, such as process metrics, source code text, socio-technical metrics, object oriented metrics, and line of code metrics [16][17][18]. In our study, we focus on existing machine learning algorithms that may lead to predicting/estimating fault incidents using provenance data.

III. APPLICATION USE SCENARIOS

In order to define the scope of the proposed research, we outline several application usage scenarios and various requirements of the desired self-healing system architecture. This section identifies several such scenarios, which differ in terms of the devices used, their number, granularity, and their interaction capabilities.

A. Elderly surveillance

This application aims at capturing important information from elderly people and sending it back to a central platform. It also serves as an agenda, reminder and telephone. Outside, it works as a global positioning system (GPS). The primary areas of application of the IoT in this scenario are shared with those in typical healthcare systems: tracking, identification and authentication, sensing and data gathering. This system works on a mobile platform, being dependent on availability of internet signal and energy. Moreover, it takes into consideration wearable sensors for acquiring vital information, which ship it to the mobile device via bluetooth, and from the device into the central, in real-time. Different sorts of services are coordinated with each other and composed to fulfill the system's functional requirements. The computational resources and battery power of these systems are limited, while communication technologies consume considerable amounts of energy. In this particular scenario, the IoT application should be capable of proactively predicting problems and should have fault tolerance. In this sense, the system should act (and react) in accordance with self-healing mechanism when detecting and predicting problems.

B. Smart Cities

The primary issue here is the way smart objects and sensors interact and are orchestrated with the families of electronic public services (EPS) that structure the urban network. A smart city is often characterized as instrumented, interconnected, and intelligent. Instrumented refers to the capacity to acquire real-world data using different types of channels like sensors, personal devices, medical devices, social networks, etc. Interconnected refers to the integration of data in an interoperable platform and its provision to and usage on different city services. Intelligent relates to the use of complex computational tools to deliver public value to city inhabitants. Due to the embeddedness of digital technology, citizens are more and more used to interacting

with them on a daily basis, typically through mobile devices and wireless networks. Therefore, cities possess a wide range of digitally skilled users that are ready to use and benefit from the IoT to deliver EPS. However, the development of smart city initiatives faces some challenges, some of them falling clearly into the domain of applications of the heterogeneous computing platforms, such as IoT. In this scenario, we argue that these challenges in developing IoT applications are rooted in the lack of self-healing capabilities associated with such IoT applications. These capabilities are very beneficial, considering the growth of connected devices, as these applications are integrating many smart environments from different domains, such as transportation, health and e-participation.

IV. SYSTEM ARCHITECTURE

In this study, we present a self-healing mechanism for IoT application domain. Inspired by our application use scenarios, we argue that given an IoT application, if some devices or services failed, IoT application would be shut down. To this end, in this study, we introduce a failover mechanism to enable fault tolerance in IoT applications, so that the application can still continue its functioning (even in the case of few failed devices/services). This failover mechanism is introduced to address the aforementioned objective#1. We present a fault prediction/estimation mechanism that could estimate the present number and future incidences of faults. We refer the failover mechanism as the Self-Healing Mechanism Component. Within this component, we also take into account both user involvement and computing environment requirements to address the objective#3. In this study, we also introduce the use of existing solutions to a Provenance Service (i.e., Metadata Service for execution traces of activities) to enable fault tolerant IoT systems. This addresses the aforementioned objective#2. Figure 1 illustrates system architecture for fault tolerance. In this section, the components and their interdependencies are explained in detail, together with the employed research methods.

A. Provenance Service

Provenance is metadata, which is defined as the lineage of a piece of data or an activity. It keeps track of the lifecycle of an activity or data. In the presented self-healing methodology, provenance metadata will be used for providing fault tolerance. To this end, PROV-O Specification (W3C recommended data representation) will be utilized for provenance data representation [19]. In provenance data representation, ideal granularity of provenance and the types of information should be considered for self-healing purposes.

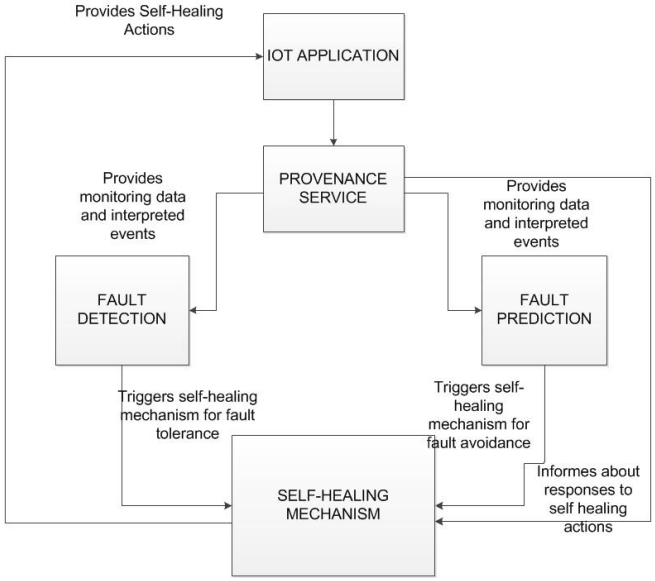

Figure 1. System Architecture

B. Fault Detection and Prediction

One of the aspects of a self-healing mechanism is to be able to protect itself from possible failures. To achieve this, we argue that the following research challenges should be taken into account.

The first challenge is data conversion. Provenance is graph-based data expressing the execution traces of activities. Since provenance data is represented in XML format, it is not suitable for data mining tasks. Distributed provenance graphs should be converted to a small-scale provenance graphs should be converted to a small-scale representation without information loss, so that they can be processed for fault prediction/estimation. Such a data conversion can be done by utilizing statistical features for performing the data conversion, without information loss for tasks like clustering of scientific workflow execution traces [20][21].

The second challenge is fault prediction/estimation. Existing machine learning algorithms that could lead to predicting/estimating fault incidents will be utilized. Within this challenge, one of the sub-goals of this study is to identify all possible faults that might occur in the aforementioned application domains. There could also be a case in which the provenance data conversion will not lead to good prediction/estimation capabilities; hence, big data processing approaches (Map/Reduce programming model) that can enable application of prediction/estimation algorithms on large-scale provenance data should be considered.

The third challenge is fault detection on runtime. To support accommodation to unexpected changes, change detection strategies should be carried out. Interdisciplinary research activities should be conducted, combining advanced data mining & knowledge discovery methodology with fault

 ISBN: 978-1-61208-474-9

detection strategies based on models including smart environment' context and human-user factors. Basic principles of fault detection imply the exploitation of redundancy in order to detect inconsistencies on real data. Such deviations are used to generate alarms associated to unexpected changes and signatures described by them are used in the identification and isolation of possible causes. Models used for this purpose can be obtained from either first principles (transient models) or learned from data (following data mining, knowledge discovery approaches). Complex event processing (CEP) has been one of the widely used method utilized to facilitate runtime fault detection for IoT. CEP is used for controlling operational rules for each device taking part in IoT separately. Here, we aim at monitoring the overall rules regarding the coordination of many systems within an IoT context.

C. Self-Healing Mechanism

In this study, we argue that self-healing systems handle fault tolerance for dynamic coordinated IoT devices taking part in IoT application. Self-healing mechanisms autonomously identify erroneous service and manage the means by which the system is repaired. Resilience is considered as a property of coordinated IoT to be deeply studied to progress towards completely automated self-healing systems. Hereby, one can consider several strategies as follows: i) a failover mechanism by providing availability to facilitate failure recovery, ii) architectural adaptation and (automated) architecture reconfiguration, iii) manufacturing values and estimations to facilitate testing of the Self-Healing Mechanism component, and iv) providing online feedback to operators in case of potential/foreseen errors. Our approach to resilience is to provide a failover mechanism. To this end, we identify following sub-components of a self-healing mechanism: a) Failover mechanism, b) Messaging protocol and messaging bus, and c) Recovery. We describe each component as follows.

Failover mechanism: We use replication to achieve fault tolerance. The technique of replication is generally used in order to increase the dependability level of data hosting environments. There are two types of replication methods: permanent replication and dynamic replication. Permanent replication stores the copies of data permanently. However, in the dynamic replication method, the copies of data are created temporarily [22][23]. In the proposed self-healing mechanism, we are interested in replicating services and providing service redundancy for fault tolerance. Employment of a combination of both permanent and dynamic replication in providing resilient IoT applications should be considered in order to provide a minimum level of replication of services (to meet with desired fault tolerance), as well as an adjustable level of replication of services (in case some services tend to be more fragile).

Messaging protocol and messaging bus: In order to achieve a decentralized replication mechanism, messaging-based replication protocols should be used. These protocols will include messages like: a) selection of replica IoT devices for replica service (both active and idle), b) selection of new active replica services, c) live-state of existing IoT devices, and d) introduction of a new IoT device into the system. The use of a topic-based publish/subscribe-based messaging paradigm, as for messaging bus, provides one-to-one, one-to-many, and many-to-one communication channels among the IoT devices. In this approach, each participating IoT device will send a ping request (liveliness information) to the rest of the available network nodes through a publish-subscribe system. Each node will keep a vector of information on existing nodes and will refresh it periodically. Whenever a fault is predicted, a self-healing system is expected to self-optimize itself for fault avoidance. Here, our approach will take inputs from the Fault Prediction mechanism and readjust the replica service configuration (e.g., selection of new active replica service, increasing the replica service numbers, etc.).

Recovery: Recovery is another aspect of a self-healing mechanism. In our self-healing mechanism approach, a recovery mechanism will include actions to provide the system with one of the idle replica services (instead of the failed service) to bring the system to a known state of replication level. Here, we intend to use messaging-based protocols for recovery as well to achieve this.

An ideal self-healing system should implement the fault-tolerance related tasks, implicitly optimizing the use of resources of the system and the involvement of users. Users must be involved in the customization of recovery or tolerance of failures in the IoT applications that they generate. We argue that the proposed approach to model replication strategy should take into account the use of resources and involvement of users in the IoT environments.

V. CONCLUSION AND FUTURE WORK

We have discussed research challanges related to fault tolerance for IoT applications running in heretogenous computing environments. We reviewed background work on fault tolerance for these applications. We explained application use scenarios to define the scope of this study.

The expected contributions of this research can be outlined as follows. This study presents a fault tolerance methodology that could address the resilience requirements of IoT applications. It defines architectural constraints for building fault tolerance in IoT application domains and proposes a self-healing mechanism for IoT application domains. This approach includes the use of replication of services and utilizes topic–based, publish-subscribe messaging protocols to achieve fault tolerance.

In the future work, we will introduce a) a failover mechanism, b) machine learning algorithms to perform forecasting/estimations, c) a methodology to define the fault tolerance related tasks. Furthermore, we also plan on manufacturing values and estimations to facilitate testing of the Self-Healing Mechanism component and providing online feedback to operators in case of potential/foreseen errors.

ACKNOWLEDGMENT

We would like to thank Software Testing and Quality Evaluation Laboratory (YTKDL) of TUBITAK-BILGEM and Software Quality Laboratory of Yildiz Technical

University for supporting us and allowing us to use their computer facilities for this study. As always, we are really grateful for the help of the extended team of our department.

REFERENCES

[1] A. Botta, W. Donato, V. Persico, and A. Pescape, "On the Integration of Cloud Computing and Internet of Things", IEEE, 2014, pp. 23-30, ISBN: 978-1-4799-4357-9.

[2] U. Yildiz, P. Mouallem, M. Vouk, D. Crawl, and I. Altintas, "Fault-Tolerance in Dataflow-based Scientific Workflow Management", IEEE, 2010, pp. 336-343, ISBN: 978-0-7695-4129-7.

[3] N. Finne, "Towards Adaptive Sensor Networks," Dissertation for the degree of Licentiate of Philosophy in Computer Science, Uppsala University, 2011.

[4] T. Bourdenas and M. Sloman, "Starfish: policy driven self-management in wireless sensor networks", Proceedings of the 2010 ICSE Workshop, 2010, pp. 75-83, ACM 978-1-60558-971-8.

[5] J. Neumann, N. Hoeller, C. Reinke, and V. Linnemann, "Redundancy Infrastructure for Service-Oriented Wireless Sensor Networks", in 9th IEEE International Symposium on Network Computing and Applications (NCA 2010), IEEE Computer Society , July 2010, pp. 269–274, ISBN: 978-0-7695-4118-1.

[6] K. Piotrowski, P. Langendoerfer, and S. Peter, "tinyDSM: A highly reliable cooperative data storage for Wireless Sensor Networks", in 2009 International Symposium on Collaborative Technologies and Systems, IEEE, 2009, pp. 225–232, ISBN: 978-1-4244-4586-8.

[7] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage," in Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02), ACM, 2002, vol. 5, pp. 78–87.

[8] P. H. Su, C. Shih, J. Y. Hsu, K. Lin, and Y. Wang, "Decentralized Fault Tolerance Mechanism for Intelligent IoT/M2M Middleware", IEEE World Forum on Internet of Things (WF-IoT), IEEE, 2015 pp. 45-50, ISBN: 978-1-4799-3459-1.

[9] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges ofsecurity and privacy in distributed internet of things", Computer Networks, vol. 57, no. 10, 2013, pp. 2266-2279.

[10] F. M. Almeida, A. R. L. Ribeiro, and E. D. Moreno, "An Architecture for Self-healing in Internet of Things", UBICOMM 2015 : The Ninth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, IARIA, 2015, pp. 76-81, ISBN: 978-1-61208-418-3.

[11] H. M. Salmon, et al.. "Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques", International journal of wireless information networks, vol. 20, no. 1, 2013, pp. 39-66.

[12] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things." Ad hoc networks, vol. 11, no. 8, 2013, pp. 2661-2674.

[13] A. P. Athreya, B. DeBruhl, and P. Tague, "Designing for Self-Configuration and Self-Adaptation in the Internet of of Things", Carnegie Mellon University, 2013, pp. 585-592.

[14] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings", IEEE Trans. Software Eng., vol. 34, no. 4, pp. 485-496, July/Aug. 2008, (Paper=97, Status=F, Phase=2, Data=N).

[15] E. Arisholm, L.C. Briand, and E.B. Johannessen, "A Systematic and Comprehensive Investigation of Methods to Build and Evaluate Fault Prediction Models", J. Systems and Software, vol. 83, no. 1, 2010, pp. 2-17. (Paper=9, Status=P)

[16] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering", IEEE Transactions On Software Engineerıng, 2012, Vol. 38, No. 6.

[17] S. Shivaji, E.J. Whitehead, R. Akella, and K. Sunghun, "Reducing Features to Improve Bug Prediction", Proc. IEEE/ACM 24th Int'l Conf. Automated Software Eng., 2009, pp. 600-604. (Paper=164, Status=P).

[18] C. Bird, N. Nagappan, H. Gall, B. Murphy, and P. Devanbu, "Putting it All Together: Using Socio-Technical Networks to Predict Failures", Proc. 20th Int'l Symp. Software Reliability Eng., 2009, pp. 109-119. (Paper=18, Status=P).

[19] PROV-DM: The PROV Data Model. [online] Available at: http://www.w3.org/TR/prov-dm/[Accessed 14 Nov. 2015].

[20] M. Aktas, B. Plale, D. Leake, and N. Mukhi, "Unmanaged Workflows: Their Provenance and Use", Data Provenance and Data Management in eScience, Berlin Heidelberg: Springer-Verlag, 2013, pp. 59-81.

[21] P. Chen, B. Plale, and M. S. Aktas, "Temporal representation for mining scientific data provenance", Future Generation Computer Systems-The International Journal Of Grid Computing And Escience, 2014, 36, pp. 363-378.

[22] M. Rabinovich, I. Rabinovich, R. Rajaraman, and A. Aggarwal, " A Dynamic Object Replication and Migration Protocol for an Internet Hosting Servic in Proc.", 19th Int'l Conf. Distributed Computing Systems, 1998, pp. 101-113.

[23] M.S. Aktas and M. Pierce, "High-performance hybrid information service architecture", 2010, Concurr. 22(15), pp. 2095-2123.

 ISBN: 978-1-61208-474-9