Future Generation Computer Systems 45 (2015) 123–132

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

On-line failure prediction in safety-critical systems☆

Roberto Baldoni [a,b], Luca Montanari [a,b,*], Marco Rizzuto [c]

[a] Cyber Intelligence and Information Security Research Center, "Sapienza" University of Rome, Italy
[b] Department of Computer, Control, and Management Engineering Antonio Ruberti, "Sapienza" University of Rome, Italy
[c] Selex-ES, Rome, Italy

HIGHLIGHTS

- Non-intrusive and black box effective online failure prediction.
- We monitor network traffic, only, to perform online failure prediction.
- Application agnostic: no knowledge of application logic is required.
- We use complex event processing to produce a representation of the system state.
- We use hidden Markov models in order to create a state recognizer.

ARTICLE INFO

Article history:
Received 29 October 2013
Received in revised form
11 November 2014
Accepted 18 November 2014
Available online 27 November 2014

Keywords:
Failure prediction
Complex event processing
Machine learning
Complex distributed systems
Critical infrastructures

ABSTRACT

In safety-critical systems such as Air Traffic Control system, SCADA systems, Railways Control Systems, there has been a rapid transition from monolithic systems to highly modular ones, using off-the-shelf hardware and software applications possibly developed by different manufactures. This shift increased the probability that a fault occurring in an application propagates to others with the risk of a failure of the entire safety-critical system. This calls for new tools for the on-line detection of anomalous behaviors of the system, predicting thus a system failure before it happens, allowing the deployment of appropriate mitigation policies.

The paper proposes a novel architecture, namely CASPER, for online failure prediction that has the distinctive features to be (i) *black-box*: no knowledge of applications internals and logic of the system is required (ii) *non-intrusive*: no status information of the components is used such as CPU or memory usage; The architecture has been implemented to predict failures in a real Air Traffic Control System. CASPER exhibits high degree of accuracy in predicting failures with low false positive rate. The experimental validation shows how operators are provided with predictions issued a few hundred of seconds before the occurrence of the failure.

1. Introduction

A few years ago, safety-critical systems traditionally used in air traffic control, commercial aircraft, nuclear power, consisted of a monolithic (possibly proprietary) system provided by a single vendor. Such systems thus incurred high cost of development and maintenance. To reduce such costs, systems have been unpacked in a set of applications/services (usually developed by different vendors) that interact through a set of well-defined interfaces.

Applications need to meet stringent Quality of Service (QoS) requirements in terms of availability in order to ensure, in their turn, the high availability of the whole safety-critical system. To achieve this objective, applications require to distribute and replicate data (e.g., flight routes in Air Traffic Control system) on a number of nodes connected through a WAN or a LAN. Due to the nature of such systems, the replicas of an application need to be strictly consistent in order that they keep the same state along the time, providing a client with the illusion that its request occurs instantaneously [1].

In such complex distributed systems, failures are a matter of life thus they have to be safely handled to ensure system survivability. Extensive testing during the design phase of an application cannot avoid the occurrence at operational time of failures that can lead to catastrophic consequences for the entire system functioning.

☆ A preliminary version of this paper appeared at SAFECOMP 2012.

* Corresponding author.
E-mail addresses: baldoni@dis.uniroma1.it (R. Baldoni), montanari@dis.uniroma1.it (L. Montanari), mrizzuto@selex-es.com (M. Rizzuto).

http://dx.doi.org/10.1016/j.future.2014.11.015
0167-739X/

Keeping a set of replicas strictly consistent in the presence of replica failures boils down indeed to solve the consensus problem [2]. Thus, if the distributed system has a good coverage of synchrony assumptions (i.e., network and computing nodes are working nicely), there are a number of fault tolerance mechanisms that can be used to overcome the failure and to keep replicas consistent (e.g., failure detection through heart-beating [3]). If a fault happens in a period when there is no coverage of synchrony assumption, replicas might manifest anomalous behaviors due to well-known Fischer–Lynch–Paterson (FLP) impossibility result stating that distributed consensus cannot be reached in an asynchronous system even in the presence of only one faulty process [4]. These behaviors could transitively affect other applications bringing in the worst case to a system failure such as, for example, an abnormal system shutdown. In this case, it might take a long time to resume a correct functioning, severely reducing system availability. The only way to avoid such abnormal system failures is to predict them, sensing the occurrence of anomalous behaviors. Accurate and timely predictions can help to mitigate the effect of failures by taking appropriate recovery actions before the failure occurs. Such actions can mitigate the loss of availability by reducing the time needed to resume a correct system behavior.

The industrial trend is to split a safety-critical system into at least two distinct sets of interconnected applications/services possibly developed by two or more distinct manufacturers. The first one handles the application logic (e.g. flight routes management in Air Traffic Control Systems) and the second set is devoted to control and supervision that, among the others, has the task of managing failures. Manufacturers require that supervision services interfere as less as possible with the application logic. This is due to the fact that application logic has a long and complex deployment phase at customer premises and the performance of the application logic cannot be influenced in any way by the supervision mechanisms. This implies that for example application logic and supervision cannot be co-located on the same nodes.

For all these reasons, some prominent future challenges of next generation safety-critical systems relate to the capacity of keeping the same, or even increase, the availability of such systems while considering this open multi-manufacturers setting. In particular:

- High-level assurance of non-interference at run-time among distinct applications forming the safety-critical distributed system. Some of these applications should not have any interaction at all (e.g., supervision and application logic) in order to ensure the mutual correctness of their behavior;
- An application, which is a part of the safety-critical distributed system, would have a great value if agnostic with respect to the internals of all others interacting applications. As an example, supervision should have zero knowledge on the deployment and on the source code of the application logic.

These challenges translate into having a supervision that acts like both a *non-intrusive* and *black-box* observer. *Non-intrusive* means the failure prediction does not use any kind of information on the status of the nodes (e.g., CPU, memory) of the monitored system; *black-box* means no knowledge of the application internals and of the application logic of the system is exploited. Operationally, in safety-critical systems, a large amount of data, deriving from communications among applications and services, transits on the network; thus, the "observer" can focus on that type of data, only, in order to recognize many aspects of the actual interactions among the components of the system.

This paper presents a first attempt in the direction of designing next generation highly available safety-critical systems by introducing a novel non-intrusive and black-box failure prediction architecture, namely CASPER. It works *online* since the failure prediction is carried out during the normal functioning of the monitored system and it exploits only information traveling on the network interconnecting nodes of the supervised and monitored systems. Specifically, the aim of CASPER is to recognize any deviation from normal behaviors of the monitored system by analyzing symptoms of faults that might occur in the form of anomalous conditions of specific performance metrics. In doing so, CASPER combines, in a novel fashion, *Complex Event Processing* and machine learning algorithms designed through *Hidden Markov Models*. The latter is used to classify at run time nice states and anomalous states of the monitored system. Roughly speaking when an anomalous state is detected an alarm is sent to the operator. In this paper we show how CASPER can be effective also implementing the simplest version of HMM. We indeed deployed CASPER for monitoring a real Air Traffic Control (ATC) system developed by Selex-ES (a Finmeccanica Company) and conducted a 6 months long experimental evaluation. Results show CASPER timely predicts failures in the presence of memory and I/O stress conditions while keeping reasonably low[1] the number of false positives.

2. Related work

A large body of research is devoted to the investigation of approaches to online failure prediction. The interested readers can find a comprehensive taxonomy of them in [5]. In this section, we discuss only those that are closer to our solution and that mainly inspired the approach we followed. We can distinguish between approaches that make use of symptoms monitoring techniques for predicting failures (as CASPER does), and approaches that use error monitoring mechanisms. In the latter category, Salfner in [6] presents an error monitoring failure prediction technique that uses Hidden Semi-Markov Model (HSMM) in order to recognize error patterns that can lead to failures. Hoffman et al. in [7] describe two non-intrusive data driven modeling approaches to error monitoring, the first based on a Discrete Time Markov Model and the second one on function approximation.

In the context of symptoms monitoring, there exist failure prediction systems that use black-box approaches, namely ALERT [8] and Tiresias [9]. Both systems are intrusive in the sense that they interfere with hosts running the application logic by sensing internal parameters such as CPU consumption, memory usage, input/output data rate. Using such internal parameters allow to quickly sense bad behaviors and alert operators. Thus, one of the challenges of CASPER has been to get good accuracy and fast response time while being non-intrusive. A more recent work [10] that can be considered non-intrusive as it uses network traffic only, presents data-mining techniques that extract essential models of anomalous behavior sequences known to be precursors of system failure conditions, i.e., symptoms in our sight. The main difference is that it is designed to address network failures only, while our work is more general in that sense. From a model point of view, in the field of failure prediction, [11] presents prediction sub-model for each component and combines them using component dependencies. This is different from other approaches as usually the same model is used for all the components. Note that [11] is intrusive as it requires the installation of monitoring probes to the observed components.

Out of the area of failure detection systems, CASPER got inspired by the following approaches: Aguilera et al. in [12] consider the problem of discovering performance bottlenecks in large scale distributed systems consisting of black-box software components. The system introduced in [12] solves the problem by using message-level traces related to the activity of the monitored system in a non-intrusive fashion (passively and without any

[1] False positive rate less then 12%, F-measure more then 80%, see Section 6.2 for details.