An online failure prediction system for private IaaS platforms

[Extended Abstract]

Pedro Capelastegui
Center for Open Middleware (COM)[†]
Universidad Politécnica de Madrid (UPM)
pedro.capelastegui@[*]

Alvaro Navas
Center for Open Middleware (COM)[†]
Universidad Politécnica de Madrid (UPM)
alvaro.navas@[*]

Francisco Huertas
Center for Open Middleware (COM)[†]
Universidad Politécnica de Madrid (UPM)
francisco.huertas@[*]

Rodrigo Garcia-Carmona
Departamento de Ingeniería de Sistemas Telemáticos (DIT)[‡]
Universidad Politécnica de Madrid (UPM)
rodrigo@dit.upm.es

Juan Carlos Dueñas
Center for Open Middleware (COM)[†]
Universidad Politécnica de Madrid (UPM)
juancarlos.duenas@[*]

ABSTRACT

The size and complexity of cloud environments make them prone to failures. The traditional approach to achieve a high dependability for these systems relies on constant monitoring. However, this method is purely reactive. A more proactive approach is provided by online failure prediction (OFP) techniques. In this paper, we describe a OFP system for private IaaS platforms, currently under development, that combines different types of data input, including monitoring information, event logs, and failure data. In addition, this system operates at both the physical and virtual planes of the cloud, taking into account the relationships between nodes and failure propagation mechanisms that are unique to cloud environments.

Categories and Subject Descriptors

C.4 [**Performance of systems**]: [Fault tolerance, Reliability, availability, and serviceability]; C.2.4 [**Computer-communication networks**]: Distributed Systems—*Distributed applications*

General Terms

Reliability, Design

Keywords

OFP, cloud, IaaS, reliability

[*]centeropenmiddleware.com
[†]Campus de Montegancedo
E-28223 Pozuelo de Alarcón, Madrid, Spain
[‡]Av. Complutense, 30
28040 Madrid, Spain

DISCCO 2013 Braga, Portugal

1. INTRODUCTION

Interest in cloud computing has been on the rise in recent years, as companies move away from traditional physical infrastructures to deploy their systems on public clouds like Amazon EC2, which offer access to cheap on-demand scalable computational resources. However, this flexibility and scalability come with a loss of control, as the physical infrastructure is owned by an external entity. Because of this, there is a growing interest in the private IaaS cloud model, which offers some of the benefits of public clouds without relinquishing the control over the physical infrastructure.

However, these private cloud solutions are significantly less evolved than public offerings, specially concerning reliability. Current systems enforce a passive approach to fault management, monitoring physical machines and setting alarm thresholds for each observed resource. In such systems, alarms are often triggered with no time to react and avoid a potential loss. In order to improve failure response time, a more proactive approach is required, such as Online Failure Prediction (OFP) techniques, which aim to predict future system failures by analysing monitoring data, error logs, or previous failures.

In this paper we present the architecture of an OFP system aimed at private IaaS platforms. The system, currently under development, analyses data from virtual machines (VMs) in a cloud as well as physical machines (PMs), and combines different prediction approaches (based on monitoring, events, and failures) to improve prediction quality. In addition, we propose using awareness of cloud topology (in both physical and virtual planes) and knowledge of failure propagation mechanisms in a cloud to further improve

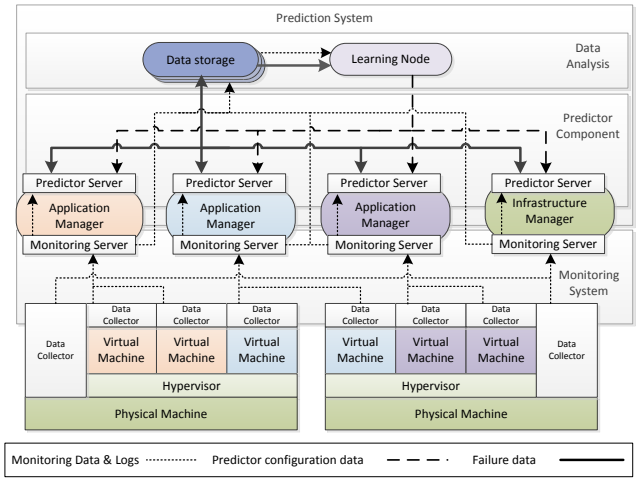

Figure 1: The multi-level system architecture

predictions.

2. RELATED WORK

A recent survey of OFP methods in general is provided in [3]. Although there is abundant prior work discussing OFP applied to large distributed systems, its use with cloud computing systems remains a relatively new field. OFP solutions for clouds [5] [4] [2] have to take into account the unique characteristics of these systems, such as their highly dynamic structure and configuration, and their reliance on virtualization. However, these solutions tend to only use data from PMs and, at most, get VM information indirectly from hypervisors. By contrast, our proposed architecture performs predictions at multiple levels (physical, virtual node, virtual application, cloud), covering a greater variety of potential failures. Furthermore, our system gains access to more detailed VM information by directly accessing monitoring and logging from each VM. Finally, while OFP solutions often focus on a single prediction approach, we are experimenting with combining multiple approaches to improve predictions.

3. PREDICTION SYSTEM

3.1 Architecture

Figure 1 shows the prediction system architecture and its integration with a monitoring system. We are using a previously existing monitoring system presented in [1], since it can provide a multi-level view of the cloud, gathering information about the PMs supporting the infrastructure, but also about cloud VMs and the cloud applications they compose, as well as the relationships between all of them. For example, we could use this system to examine a specific application running on the cloud, the VMs within that application, and the PMs hosting these VMs. This gathering approach is aimed at private IaaS platforms, as it would be too invasive for public clouds, and too low-level for a PaaS.

The monitoring system defines an Infrastructure Manager (IM) to handle data from cloud PMs, and several Application Managers (AMs) for the data from a given cloud application and its associated VMs. Every machine in the cloud, whether physical or virtual, includes a Data Collector that measures resource usage and collects application logs. This data is sent to a Monitoring Server, which processes it (e.g. filtering and aggregating it) and incorporates it to its system model. Each Monitoring Server is assigned to a IM or AM, as appropriate.

The prediction system comprises two parts: the predictor component, which makes failure predictions based on semi-real-time data, and the data analysis cluster, which stores long term data gathered by the monitoring system and analyses it in order to tune the prediction algorithms. This distinction allows us to lighten the predictor so that it can be deployed on any kind of machine and ran in real time, while simplifying the aggregation of data from several sources and enabling the study of data over longer time periods.

The predictor component is made up of several Predictor Servers, one of which is assigned to the IM and to each AM. On each server, multiple prediction techniques are used and combined, as we explain in the following section. A server takes monitoring data and logs from its neighbour Monitoring Server, and system failure data from other predictor servers (transmitted through the management network between IM and AMs), and uses that data to perform predictions. On a predicted failure, the predictor component emits an alarm associated to a PM, VM, or cloud application, warning of an expected failure within a certain prediction period.

Each predictor server only keeps data corresponding to a limited time period. However, all monitoring, logging, and failure data is periodically processed and sent to a storage unit at the data analysis cluster for long term storage and analysis. That data is then used by the Learning Node, which updates ever day each prediction algorithm in each Predictor Server. The details of this update process will depend on each specific algorithm, but we can summarize it as a supervised learning process where failure data is combined with a training dataset to generate an optimal set of parameters for a given algorithm and prediction domain. These updated algorithm parameters are stored and, eventually, distributed to their corresponding Predictor Servers. It is important to note that, since a cloud can contain hundreds of physical machines and even more VMs, storing and processing all this data requires the use of Big Data techniques. We have decided to use a Hadoop cluster to fulfil this need due to its flexibility and open-source nature.

Finally, we have to point out that this is still a work in progress. The monitoring system is already implemented but we are currently working on the predictor component and analysis cluster. Once we have a working prototype, we will be able to properly evaluate the efficiency of this approach.

3.2 Prediction approach

Our failure prediction approach operates on two axes: the virtual plane of the cloud (VMs running on the cloud), and the physical plane (physical cloud servers on which VMs run). Each of these planes presents different usage and failure patterns, and may have different types of data available for prediction analysis. Since some failures can only be predicted from a specific cloud plane (e.g. application-level failures that are observable on VM logs, but transparent to the physical machine), we want our system to predict on both planes.

On the other hand, we have three categories of failure

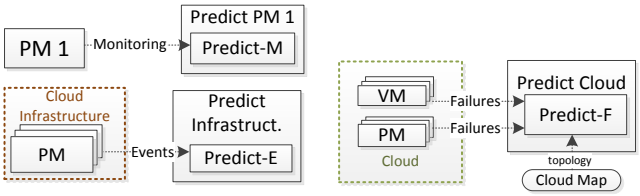

Figure 2: Predictor Server in Infrastructure Manager

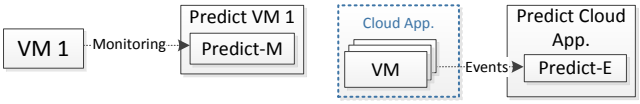

Figure 3: Predictor Server in Application Manager

prediction mechanisms, based on the choice of input data: monitoring based prediction, which periodically examines monitored system variables to predict resource exhaustion, event-based prediction, which searches for patterns preceding failures in event logs, and failure-based prediction, which analyses temporal and spatial distribution of failure occurrences. Each of these categories excels at identifying different types of failures, and there are failures that can only be anticipated with a specific type of predictor, so we want our system to support all three.

It is also important to consider the relationships between different cloud elements, and their impact on error occurrence. For the purposes of our analysis, we have identified two main relationships: the hosting relationship between a VM and its corresponding PM, and the common execution context shared between all VM within a cloud application. We have noticed that, through these relationships, system failures can propagate between cloud elements, in the following ways:

- A failure in a PM can cause failures in its hosted VMs. On a related note, a VM failure can be a symptom of an upcoming failure in its corresponding PM.
- A failure in a VM can produce further failures in other VMs within the same cloud application. This can be due to network failures propagating through a virtual network, or application level failures affecting several nodes working together.
- As with non-cloud systems, a failure in a PM can propagate to nearby PMs. Examples of this include failures due to ambient conditions, and network failures.

Figures 2 and 3 show the proposed structure for our predictor, with separate predictor servers in the IM and AMs, respectively. This covers failures in both the physical and virtual planes of the cloud, and exploits all three prediction mechanisms, while factoring the potential for failure propagation between cloud elements. Prediction based on monitoring (Predict-M) is performed for each individual machine, whether virtual or physical, whereas event-based prediction (Predict-E) processes aggregations of log events for all VMs in each cloud application and all physical machines composing the cloud infrastructure, respectively. Failure-based prediction (Predict-F) uses all cloud failure data alongside a cloud map indicating physical and logical distance between virtual and physical cloud nodes.

To illustrate the usefulness of this approach, consider a scenario where a cloud application had a bugged database. The first time the bug was observed, there would be a failure at the database VM, and the prediction system would alert about potential failures in other VMs within that application. Afterwards, the system could learn to predict that kind of failure by analysing event logs from database VMs.

4. CHALLENGES AND RISKS

Event-based prediction over a cloud application presents the challenge of not knowing in advance the structure of event logs, as they depend on the specific software and configuration selected by cloud users. This can be addressed by using a format-agnostic prediction mechanism, like the one proposed in [5]. On the other hand, event-based predictors require a significant training time to achieve optimal prediction accuracy, but cloud application life time shows great variability, from a few hours to months. As a consequence, the shorter-lived subset of cloud applications may not operate long enough to benefit from this technique.

The propagation of failures between cloud VMs and from PMs to VMs is, at this stage, a theory that we have yet to validate through practical experimentation. Hence, we still do not know the magnitude of the correlation between failures, and whether it will allow for a useful predictor.

5. ACKNOWLEDGEMENTS

The work for this paper was partially supported by funding from ISBAN and PRODUBAN, under the Center for Open Middleware initiative.

6. REFERENCES

[1] R. Garcia-Carmona, F. Cuadrado, A. Navas, A. Celorio, and J. C. Dueñas. A multi-level monitoring approach for the dynamic management of private iaas platforms. *Journal of Internet Technology*, Accepted for publication.

[2] Q. Guan, Z. Zhang, and S. Fu. Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems. *Journal of Communications*, 7(1):52–61, 2012.

[3] F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. *ACM Computing Surveys (CSUR)*, 42(3):10, 2010.

[4] C. Wang, V. Talwar, K. Schwan, and P. Ranganathan. Online detection of utility cloud anomalies using metric distributions. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 96–103, 2010.

[5] Y. Watanabe, H. Otsuka, M. Sonoda, S. Kikuchi, and Y. Matsumoto. Online failure prediction in cloud datacenters by real-time message pattern learning. In *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 504–511, 2012.