MonArch Multilayer Monitoring and Analytics in Software-Defined Infrastructures

Jieyu Lin
University of Toronto
Toronto, Ontario
jieyu.lin@utoronto.ca

Hadi Bannazadeh
University of Toronto
Toronto, Ontario
hadi.bannazadeh@utoronto.ca

Alberto Leon-Garcia
University of Toronto
Toronto, Ontario
alberto.leongarcia@utoronto.ca

ABSTRACT

Monitoring is crucial for enabling intelligent management in today’s virtualized and heterogeneous IT infrastructure. Flexible scalable analytics and Monitoring-as-a-Service (MaaS) are two key features of next generation monitoring systems. To meet these requirements, we designed and implemented MonArch, a cross-layer MaaS system that provides collection, storage, and analytics capabilities. This demo shows MonArch’s flexible multi-layer analytics by: 1) creating and displaying the communication graphs among VMs in a data center, and 2) identifying anomalies when an application running in the VMs is attacked.

1. MONITORING AS A SERVICE

Cloud computing infrastructures are growing not only in size but also becoming increasingly heterogeneous. The management system must deal not only with computing and networking, but also GPUs, programmable hardware (FPGAs), and other resources. Large volumes of monitoring data are generated from data sources in different layers of the infrastructure. To efficiently utilize these data, the monitoring system should provide flexible storage and analytics, in addition to the collection and visualization functionalities provided by existing systems.

MaaS is the key to providing an open monitoring environment that can support the needs of multiple tenants in shared resource environments. MaaS requires that the monitoring system be responsible for data collection, storage and analysis. The system can then provide open APIs to allow users to: 1) stream/submit custom monitoring data to the system for storage and analysis purposes; 2) submit analytics tasks and retrieve the results on demand; 3) access historical monitoring data on demand (with access control). The monitoring system can leverage its large volumes of cross-layer, heterogeneous monitoring data, to conduct infrastructure-wide analytics that benefit its tenants. For example, such analytics can include security detection and auditing, dynamic resource allocation, and automatic system diagnosis.

The challenges of designing a monitoring system that meets the requirements above are: 1) Scalability: scale up the system as the number of users and resources increases; 2) Heterogeneity: (structured / unstructured) monitoring data format from heterogeneous data sources (e.g. compute, network, reconfigurable device i.e. FPGA, GPU, access point); 3) Multiple-layer infrastructure: data centers contain multiple layers: physical, virtual, service, and application layers; 4) Complex and dynamic system behaviour.

2. MONITORING SYSTEM ARCHITECTURE AND IMPLEMENTATION

Based on the aforementioned requirements, we designed and implemented the MonArch monitoring system [2]. The architecture (shown in Figure 1) has four layers: 1) acquisition layer for data collection, 2) transportation layer for transporting and queuing monitoring data, 3) processing layer to provide stream and batch processing capabilities; and 4) user access layer to allow users to access monitoring data and analytics results.

Figure 1: Monitoring and Measurement System Architecture

Our implementation provides the required functionalities while emphasizing scalability. For data collection, we use Ceilometer to monitor the OpenStack virtualization layer. We implemented user agents for application layer monitoring, for OpenFlow SDN monitoring, and for physical server and hardware monitoring. We use the HBase NoSQL database for storage. We built our batch and stream analytics on top of Apache Spark. This allows us to leverage the scalability and fault tolerance that is built in Spark. Since the system’s storage (NoSQL) and analytics (Spark) components are natively scalable, the multiple data collection engine provided by Ceilometer enables high scalability. The only remaining scaling bottleneck is the network bandwidth. We can solve this issue by offloading simple analytics tasks closer to the data collection point or using a hierarchical architecture.

3. DEMO

MonArch is deployed in the SAVI testbed [1], a multi-tiered cloud infrastructure spanning Canada that was designed and implemented based on the concept of Software Defined In-
In the demo, we will demonstrate MonArch’s monitoring and analytics capability through the MonArch dashboard (shown in Figure 2). We will use MonArch to create a communication graph (heat map) of the VMs in a data center (shown in Figure 3) by using OpenFlow monitoring data and Ceilometer monitoring data. In the figure, the black-border nodes are virtual machines and the blue-border nodes are applications running on top of virtual machines. The fill color of the VM node represents the CPU utilization (red denotes higher utilization). The links in this figure are OpenFlow-based links. They are generated by processing OpenFlow monitoring data. The color of each link represents its bandwidth utilization. In this figure, we can see many different kinds of applications running in the SAVI testbed. Some examples of these applications include Hadoop clusters (mesh network with high link bandwidth usage), clusters with a load balancer (star topology), network experiment clusters.

One of the analytics capabilities of MonArch is anomaly detection. For demonstration purposes, we will launch a low and slow denial of service attack at an Apache web server running in a virtual machine. Since MonArch is continuously performing stream processing with monitoring data from physical, virtual and application layer for anomaly detection purposes, we will show how it can detect an anomaly caused by the attack. MonArch will then analyze monitoring data to show the communication graph of the abnormal VM cluster before and after the anomaly is detected (shown in Figure 4). The color difference (i.e. utilization different) of the two graphs can help us determine that the cause of this anomaly may be an attack.

4. CONCLUSION & FUTURE WORK
MonArch is a monitoring system that provides flexible and scalable analytics and Monitoring-as-a-Service. It is capable of performing monitoring data collection, storage, and analytics. MonArch’s analytics and anomaly detection capabilities are shown in this demo. The next step in our work is to provide root cause analysis capability in our system by analyzing anomalies detected in the system and correlating monitoring data from multiple resources in different layers. We are also planning to improve the system to support integrated multi data center monitoring and analytics.

5. REFERENCES