CHEP 2018

9-13 July 2018 - Sofia, Bulgaria

Developing a monitoring system for Cloud-based distributed data-centers

G. Vino , M. Antonacci, G. Donvito, D. Elia

INFN Sezione di Bari – Italy

Introduction

Nowadays more and more data-centers cooperate with each others to achieve a common and more complex goal, then non-conventional monitoring tools are needed. The proposed solution provides **proactive support** to the management of a data-center cluster. **Specific machine learning algorithms** executed on the collected monitored data **classify**, **organize** and **estimate** the status of each component in order to detect the **preliminary phases** of known malfunctions and to provide the active support to speed up the **recovery** activities.

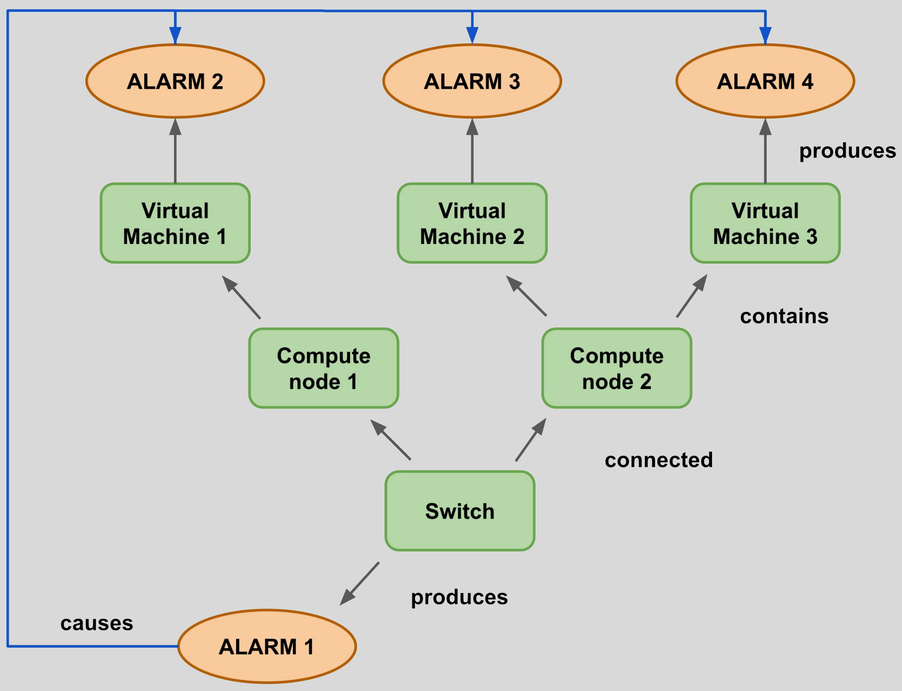
Monitoring Data Selection

Many efforts have been put into the selection of the information to be gathered from the monitored items since the quality of the results produced by the machine learning algorithms strictly depends on the input monitoring information. Heterogeneous and fully informative data have been collected for the following source categories:

- **Services** (HTTP Servers, DBs).
- **Middleware** and **IaaS layer** based on OpenStack.
- **Hardware layer** (physical servers, network devices, disk controllers).

Root Cause Analysis

The active support to the management has been implemented by the **Root-Cause Analysis engine**: it aims at finding the root-cause among a set of errors belonging to the same network of hosts. It is composed of an *event extractor*, a *graph builder* and *root-cause extractor*. The **event extractor algorithm** is an incremental *Self-Organizing Map* that associates points with similar behavior to the same label and marks each label with a critical score: a point belonging to a high critical score label generates an event. The **graph builder** extracts relationships from the monitoring data in order to create the *dependency graph*. The **root-cause extractor** merges the results produced by the previous components, creates the *cause-effect event graph* and extracts the root-cause. Currently tests and interaction with experts are ongoing for parameter tuning. Advances features like integration with the **ticketing system** and the execution of specific **actions** are currently being evaluated.



Architecture

Monitoring data about services, virtual machines and hardware devices are extracted from **Zabbix**, the local monitoring system; this information is integrated with the data provided by **Ceilometer** and the data extracted with custom scripts that use **Openstack APIs**. Others selected data sources are local batch system, **HTCondor**, and the **kernel and services logs**. All monitoring data are sent to **Apache Kafka** that offers buffering and decoupling features and, thus, increases the system availability. The data are then retrieved from **Apache Flume** agents and transmitted to the selected backends: **HDFS** (long-term archive), **InfluxDB** (timeseries database, ables to aggregate data on-the-fly) and **Elasticsearch** (search engine used to extract statistics from logs). **Grafana** and **Kibana** are used to plot timeseries and log data within dashboards. **Apache Spark** is the tool selected for the execution of all the implemented algorithms due its unified analytics engine and its capability to build easily scalable fault-tolerant applications. It executes the algorithms and interacts with **Neo4j**, a graph database. **Riemann** has been selected to send alerts. **Apache Mesos** has been deployed to increase the resource utilization efficiency. As testbed for the project the **ReCaS Bari data-center** has been used. The data-center includes the Tier2 for the ALICE and CMS experiments at CERN and hosts up to 13,000 cores, 10 PB of disk storage and 2.5 PB tape library. Actually, 22 virtual machines with 80 CPUs, 150GB RAM and 3TB of disk storage are employed.