Research Motivation

The Cloud is a complex, heterogeneous environment consisting of many layered components with complicated interconnections and relationships. Monitoring the Cloud is vital in providing service assurance and upholding SLAs. Failure to do either would result in a negative impact to customer satisfaction, potentially resulting in a loss of revenue. While the Cloud offers many advantages and interesting features (easy deployment of applications, resiliency, security, performance, scalability, elasticity, etc.), there are many challenges involved in monitoring such complex systems (Adinarayan, 2012; Aceto, 2012). Some of the main issues for monitoring the Cloud include: heterogeneity of hardware/software, scalability, aggregating and analyzing data.

Data centers, at the crux of the Cloud paradigm, are often very large buildings that house many thousands of different hardware devices. Additionally, the virtual machines that form a layer in the Cloud stack may have a wide range of operating system software, introducing yet another layer of complex heterogeneity. The different hardware/software results in a plethora of metrics and formats, leading to a significant challenge in monitoring the Cloud – aggregation of monitoring data. Cloud monitoring data is ‘big data’, with many different types of metrics calculated and recorded at frequent intervals. An important question is then, how can this information be aggregated, correlated, and analysed in an efficient and intelligent way, to make smart reconfiguration decisions? The scalability of such a solution also poses significant challenges. Given the vast number of machines (physical and virtual) and applications involved, and the many performance metrics of interest, it is vital that the information is collected and processed in a scalable way.

Literature Review

There is quite a number of monitoring tools in the Cloud arena, however, many of them don’t support reconfiguration or consider scalability. The work in (Anand, 2012) presents a scalable framework for monitoring Cloud applications, with support for cluster node monitoring. Mohamed et. al. (Mohamed 2013) propose a framework that adds monitoring and reconfiguration facilities to services of a service-based
application and deploys them in the Cloud using a scalable micro-container (Yangui, 2011). The article in (de Chaves, 2011) describes the authors’ experience with a private cloud, and discusses the design and implementation of a private cloud monitoring system (PCMONS). The authors claim that main factor impacting the success of Cloud monitoring tools is the lack of open source tools; this one of the limitations addressed by CloudMon.

**Contributions**

CloudMon is a proposed open standard for performance monitoring to overcome the heterogeneity of the Cloud stack. It has a hierarchical architecture consisting of a centralized controller and platform independent software agents (installed on every monitoring node - Fig 1). A standard set of messages is provided for metric request and reply functions to facilitate communication between the controller and agents. The standard will be extensible, and include a framework that can be used for adding new metrics e.g., new service-specific Key Performance Indicators (KPIs) can be developed to provide a better view of system performance for individual services. New scalability algorithms are needed to determine the necessary frequency of metrics and density of sensor nodes (that provide a representative sample set), given the size of the system and the required accuracy (defined in the SLA).

**Methodology**

CloudMon will be initially implemented on a small-scale testbed to validate the fundamental idea, Key Performance Indicators (KPIs), aggregation method, and scalability algorithm. The CloudMon controller (Fig 2) will support the (re)configuration of the Cloud monitoring service. It will have a web interface, where users can view the entire CloudMon network (consisting of all CloudMon enabled devices). Metrics can be enabled/disabled, and thresholds configured for each layer; and values can be viewed in the results display. The controller can be extended with logic to make intelligent VM migration decisions based on customer-defined requirements (e.g. economic, legislative, data protection). Platform independent, lightweight, software agents can be deployed on each sensor in the system. Their responsibilities include collecting, filtering, calculating new metrics, formatting and transmitting reports to the CloudMon controller.
Acknowledgement
This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie)

References

Figures

Figure 1: CloudMon Reporting Architecture

Figure 2: CloudMon Controller Flow Chart