Summary of Monitoring Tools Cloud Applications on BigData

Divya Byri
Department of CSE

ABSTRACT: Using Big Data technologies implies an unavoidable step of accumulating data which most of the time is composed on migrating the legacy database schema and facts into a relational database. However, with a view to efficiently create, check and installation new algorithms or frameworks one wishes additionally suitable tracking answers. In this paper we aim on creating a crucial review for some of the most essential monitoring solutions present on the market. Besides that we additionally gift the applicable metrics used for monitoring the cloud as well as massive records packages, with the focus on cloud deployment scenarios for big facts frameworks.

KEYWORDS—Big Data, Service Level Agreement (SLA), Software as a Service (SaaS), Platform as a Service (PaaS).

I. INTRODUCTION

The “Big data” phenomenon is now found in each quarter and function of the global economic system. Contemporary collaboration settings are often associated with huge, ever-increasing quantity of multiple forms of records, which range in terms of relevance, subjectivity and importance. Extracted knowledge can also variety from character reviews to extensively time-honored practices. Today’s agencies face demanding situations now not most effective in statistics control but in huge statistics analysis, which requires new techniques to attain insights from rather targeted, contextualised, and wealthy contents. In such settings, collaborative sensemaking very frequently take location, orchestrated or in any other case, prior to actions or selection making [34]. However, our information on how these equipment may additionally interact with users to foster and exploit synergy among human and device intelligence pretty frequently lags at the back of the technologies. The time period “information analytics” is regularly used to cover any information-pushed decisionmaking. A important funding in huge information, properly directed, can end result now not best inessential medical advances, but also lay the foundation for the next era of advances in technological know-how, medicine, and business [1].

To assist choice making, data analysts pick informative metrics that can be computed from available information with the necessary algorithms or tools, and file the effects in a way the selectionmakers can understand and act upon. Big information analytics is a workflow that distils terabytes of low-price data (e.g., every tweet) all the way down to, in a few cases, a single bit of excessive-value information (e.g., ought to Company X collect Company Y?) [5].

Technologies such as information mining, gadget mastering and semantic net are being exploited to build infrastructures and superior algorithms or services for big records analytics. Most of the services and algorithms are constructed in a generation-pushed manner with little enter from customers to power the improvement of the solutions. This can be due to: (1) customers commonly have few ideas approximately how the emerging technologies can support them; (2) issues described through users are quite preferred, such as “records overload”, “facts silos anywhere” or “loss of holistic view”, and (3) goals set by means of customers are regularly uncertain, including “locate some thing treasured”, “get an impression”, or “attain deep understandings”. It is tough to comply with conventional method of accumulating consumer necessities to lead solution development the usage of emerging technology. Another method will be a generation-driven one, i.e., how to make the technology enhance consumer’s paintings practice. However, given a numerous set of enterprise analytics state of affairs and the fact that increasingly more analytics algorithms are advanced, it’s miles hard to leverage the strengths and barriers of Big Datatechnology and follow them in specific domains [15].
This paper emphasizes on recognizing key characteristics of a monitoring solution design for the DICE project which goals to deal model-driven engineering accessible to big data application developer through an automated tool chain.

II. RELATED WORKS

Big Data is at the forefront of current distributed system research. It is geared towards data analytics on a never before seen scale. This in turn means that monitoring cloud deployments of big data frameworks is of paramount importance both for providers and consumers of such services. Firstly, it is important to monitor key performance indicators (KPI) of both the platform and application. Secondly, metrics related to Quality of Service (QoS) and Service Level Agreement (SLA) supply both the Provider and Consumer with metrics related to the overall quality and usability [19].

III. APPROACHES

In this paper we focus mainly on monitoring solutions related to big data platforms. In particular we want to highlight the performance monitoring and how this can be used to fine tune a particular deployment.

A. Monitoring Architecture

On a cloud based deployment of big data platforms cross layer monitoring is a key factor. Application components can be distributed not only on different Virtual Machines (VMs) but also on different cloud layers.

In these cases the monitoring parameters should cut across all cloud layers on which application components are deployed in order to give a complete picture of the current application status. Typically, application are deployed on one or more of the following layers: Software as a Service (SaaS), Platform as a Service (PaaS) and/or Infrastructure as a Service (IaaS). On IaaS typically we want to monitor resource utilizations such as CPU usage and states, Hard Disk utilization, Memory usage and status as well as additional network parameters. In contrast at Paas and SaaS level parameters include byte throughput metrics, status of system services, uptime, availability etc. For example, in the case of a Hadoop deployment we have metrics such as MapReduce processing time, Job Turnaround, Shuffle operations etc. The type of resources that are monitored is highly dependant on the application type. For example data transfer quality and rate is important for any video streaming application while a batch processing application will only care about basic process and network latencies.

There are several types of monitoring solutions currently in use or in development. In the case of centralized monitoring, all resource states and metrics are sent to a centralized monitoring server. These metrics are continuously pulled from each monitored component. It is easy to see that this approach while allowing a more controlled management of any cloud application has several drawbacks. First, it has a single point of fail over and lacks scalability. This means that at a certain stage the monitored application will exceed the capability of the central monitoring server and the only solution in case of centralized monitoring is that of vertical scaling. Moreover high network traffic can also lead to bottlenecks which in turn can lead to faulty or incomplete monitoring data. A decentralized approach can alleviate these problems.

On a cloud based totally deployment of big data platforms cross layer tracking is a key article. Application components may be dispensed now not only on extraordinary Virtual Machines (VMs) but additionally on unique cloud layers. In these cases
the monitoring parameters should reduce across all cloud layers on which software components are deployed that allows us to provide a whole snap of the cutting-edge application reputation. Typically software are deployed on one or extra of the following layers: Software as a Service (SaaS), Platform as a Service (PaaS) and/or Infrastructure as a Service (IaaS). On IaaS typically we need to reveal aid utilization consisting of CPU utilization and states, Hard Disk usage, Memory utilization and status in addition to extra network parameters. In comparison at PaaS and SaaS stage parameters include byte throughput metrics, repute of gadget services, uptime, availability and many others. For example, inside the case of a Hadoop deployment we have metrics together with MapReduce processing time, Job Turnaround, Shuffle operations and so forth.

The type of resources which can be monitored is particularly dependant on the software kind. For example facts switch great and change is essential for any video streaming application while a batch processing application will best care about fundamental procedure and network latencies. There are numerous kinds of monitoring solutions currently in use or in improvement. In the case of centralized tracking, all useful resource states and metrics are sent to a centralized monitoring server. These metrics are continuously pulled from every monitored component. It is straightforward to see that this technique while permitting a extra managed management of any cloud application has several drawbacks. First, it has an unmarried point of fail over and lacks scalability. This approach that at a positive stage the monitored application will exceed the functionality of the important tracking server and the best solution in case of centralized monitoring is that of vertical scaling. Moreover excessive network site visitors also can cause bottlenecks which in flip can lead to defective or incomplete tracking data. A decentralized approach can alleviate those problems.

Subsequent we will feature some of the most used monitoring tools and platforms in the context of cloud computing and big data. Some of these platforms have been assumed from HPC scenarios while others have been designed specifically for this task. Hadoop Performance Monitoring UI [7] provides an Hadoop inbuilt solution for rapidly finding performance bottlenecks and provide a visual representation of the configuration parameters which might be tuned for better performance. Fundamentally it is a lightweight monitoring UI for Hadoop server. One of its main advantages is the availability in the Hadoop distribution and the ease of usage. On the other hand it proves to be fairly limited with regard to performance. For example, the time spent in gc by each of the tasks is fairly high.

SequenceIQ [17] provides a solution for monitoring Hadoop clusters. The architecture proposed in [6] and used in order to do monitoring is based on Elasticsearch [4], Kibana [9] and Logstash [10]. The architecture proposed by [6] has the main objective of obtaining a clear separation between monitoring tools and some existing Hadoop deployment. For achieving this they use three Docker containers. In a nutshell the monitoring solution consist of client and server containers. The server container takes care of the actual monitoring tools. In this particular deployment it contains Kibana for visualization and Elasticsearch for consolidation of the monitoring metrics. Through the capabilities of Elasticsearch one can horizontally scale and cluster multiple monitoring components. The client container contains the actual deployment of the tools that have to be monitored. In this particular instance it contains Logstash, Hadoop and the collectd module. The Logstash connects to Elasticsearch cluster as client and stores the processed and transformed metrics data there.

Hadoop Vaidya [8] (Vaidya in Sanskrit language means "onewho knows", or "a physician") is a rule based performance diagnostic tool for MapReduce...
jobs. The mechanism behind Vaidya is to perform post analysis steps for map-reduce jobs. For this purpose it parses and collects different execution statistics from job history and different configuration files.

Ganglia [5], is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. The main target for Ganglia is federation formation of clusters and is based on a hierarchical design. Ganglia heavily relies on technologies as XML for data representation, XDR for data transport as well as RRDtool that is used for storing data as well as data visualization. Due to its design it manages to achieve very low per-node overheads as well as high concurrency. The tool is designed such that it is robust and easy to port to different operating systems. While developed it was ported to a vast set of operating systems and processor architecture. Currently it is in use for thousands of clusters around the world. One of its major pluses is the capability to scale up in order to handle clusters that consists of thousand of nodes. Currently being used to connect clusters from different university campuses around the world.

The Apache Ambari [1], is yet another tool that aims at making Hadoop management simpler. Ambari project is developing software that for different tasks as provisioning, managing and monitoring Apache Hadoop clusters. At its core, it also provides an easy to use and intuitive Hadoop management web user interface through RESTful APIs.

Apache Chukwa [2] is an open source data collection system for monitoring large distributed systems. Chukwa is built on top of the Hadoop Distributed File System (HDFS) and Map/Reduce framework. Since it uses these technologies it is easily scalable and robust. Besides collecting the monitoring data, it also provides a powerful tool that allows users to monitor, display and analyze results of different runs inorder to better understand the collected data. The tool is released under Apache 2.0 licence.

Datastax [3] provide a solution, OpsCenter [16], that can be integrated in order to monitor Cassandra installation. Using OpsCenter on can monitor different parameters of the Cassandra instance and also different parameters provided by the actual machines on which it runs. Also, OpsCenter exposes an interactive web UI that allow administrators to add/remove nodes from the deployment. An interesting feature provided by the OpsCenter is the automatic load balancing. For integration of OpsCenter with other tools and services an developer API is provided.

B. Monitoring requirements
In the DICE project there is a need for a monitoring and data warehousing solution. It must be able to collect and serve monitoring data from a variety of big data frameworks. Its main function is to aggregate data and to serve it to a variety of different DICE tools. In essence the monitoring solution has to be able to collect monitoring data across multiple cloud layers. In other cases, that requires a certain degree of autonomous behavior is when an already deployed big data platform has to be monitored.

In this case there has to be a mechanism that allows the discovery of all the running services in the deployment and of the underlying hardware (or VM whichever the case). This can be done using a software agent which once uploaded into a target VM (i.e. via SSH). This software is able to detect all running services and forward relevant performance metrics and logs to the logstash server.

IV. CONCLUSION
In this paper we have temporarily defined current cloud computing and big data frameworks monitoring challenges and available platforms. We have emphasized which open research queries are of principal importance in the monitoring solution that will be implemented for the DICE project. Specifically we have recognized that scaling, autonomy and timeliness are the key challenges which we have to challenge during the design and implementation of the DICE monitoring and data warehousing solution.
REFERENCES


