International Journal of Research
Available at
https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

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 associatedwith huge, ever-increasing quantity of multiple forms of records, which range in termsof relevance, subjectivity and importance. Extracted knowledge can also variety fromcharacter 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 newtechniques to attain insights from rather targeted, contextualised, and wealthycontents. In such settings, collaborative sensemaking very frequently take location,orchestrated or in any other case, prior to actions or selection making [34]. However, ourinformation on how these equipment may additionally interact with users to foster and exploit asynergy among human and device intelligence pretty frequently lags at the back of thetechnologies.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 ofadvances in technological know-how, medicine, and business [1]. To assist choice making, dataanalysts pick informative metrics that can be computed from available information withthe 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 distilsterabytes of low-price data (e.G., every tweet) all the way down to, in a few cases, a single bitof excessive-value information (e.G., ought to Company X collect Company Y?) [5].

Technologies such as information mining, gadget mastering and semantic net arebeing exploited to build infrastructures and superior algorithms or services for bigrecords analytics. Most of the services and algorithms are constructed in a generation-pushedmanner with little enter from customers to power the improvement of the solutions. Thiscan 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, suchas ‘‘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 withconventional method of accumulating consumer necessities to lead solution developmentthe usage of emerging technology.Another method will be a generation-driven one, i.e., how to make thetechnology enhance consumer’s paintings practice. However, given a numerous set of enterpriseanalytics state of affairs and the fact that increasingly more analytics algorithms areadvanced, it's miles hard to leverage the strengths and barriers of Big Datatechnology and follow them in specific domains [15].

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

This paper emphases on recognizing key characteristics of amonitoring solution design for the DICE project which goalsto 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 systemresearch. It is geared towards data analytics on a never before seen scale. This in turn means that monitoring clouddeployments 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 LevelAgreement (SLA) supply both the Provider and Consumerwith 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 usedto fine tune a particular deployment.

A. Monitoring Architecture

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

Fig.1 Monitoring Architecture Applications

In these casesthe monitoring parameters should cut across all cloud layerson which application components are deployed in order togive a complete picture of the current application status.Typically application are deployed on one or more of thefollowing layers: Software as a Service (SaaS), Platform asa Service (PaaS) and/or Infrastructure as a Service (IaaS).On IaaS typically we want to monitor resource utilizationsuch 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 processingtime, Job Turnaround, Shuffle operations etc.The type of resources that are monitored is highly dependanton the application type. For example data transfer qualityand rate is important for any video streaming applicationwhile a batch processing application will only care aboutbasic process and network latencies.

There are several types of monitoring solutions currently inuse or in development. In the case of centralized monitoring, all resource states and metrics are sent to a centralizedmonitoring server. These metrics are continuously pulledfrom each monitored component. It is easy to see that thisapproach while allowing a more controlled management ofany cloud application has several drawbacks. First, it hasa single point of fail over and lacks scalability. This meansthat at a certain stage the monitored application will exceedthe capability of the central monitoring server and the onlysolution in case of centralized monitoring is that of vertical scaling. Moreover high network traffic can also lead tobottlenecks which in turn can lead to faulty or incompletemonitoring data. A decentralized approach can alleviatethese problems.

On a cloud based totally deployment of big data platforms cross layer tracking is a key article. Application componentsmay 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 layerson which software components are deployed that allows us toprovide a whole snap of the cutting-edge application reputation.Typically software are deployed on one or extra of thefollowing layers: Software as a Service (SaaS), Platform asa Service (PaaS) and/or Infrastructure as a Service (IaaS).On IaaS typically we need to reveal aid utilizationconsisting 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 includebyte 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 dependanton the software kind. For example facts switch greatand charge is essential for any video streaming applicationwhile a batch processing application will best care aboutfundamental procedure and network latencies.There are numerous kinds of monitoring solutions currently inuse or in improvement. In the case of centralized tracking, all useful resource states and metrics are sent to a centralizedmonitoring server. These metrics are continuously pulledfrom every monitored component. It is straightforward to see that thistechnique while permitting a extra managed management ofany cloud application has several drawbacks. First, it hasa unmarried point of fail over and lacks scalability. This approachthat at a positive stage the monitored application will exceedthe functionality of the important tracking server and the bestsolution in case of centralized monitoring is that of vertical scaling. Moreover excessive network site visitors also can causebottlenecks which in flip can lead to defective or incompletetracking data. A decentralized approach can alleviatethose problems.

Subsequent we will feature some of the most used monitoring toolsand platforms in the context of cloud computing and bigdata. Some of these platforms have been assumed from HPCscenarios while others have been designed specifically for thistask.Hadoop Performance Monitoring UI [7] provides anHadoopinbuilt solution for rapidly finding performance bottlenecksand provide a visual representation of the configuration parameters which might be tuned for better performance. Fundamentallyit is a lightweight monitoring UI for Hadoop server. One ofits main advantagest is the availability in the Hadoop distribution and the eas of usage. On the other hand it proves tobe 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 Hadoopclusters. The architecture proposed in [6] and used in orderto do monitoring is based on Elasticsearch [4], Kibana [9]and Logstash [10]The architecture proposed by [6] has the main objective ofobtaining a clear separation between monitoring tools andsome existing Hadoop deployment. For achieving this theyuse three Docker containers.

In a nutshell the monitoring solution consist of client andserver containers. The server container takes care of theactual monitoring tools. In this particular deployment itcontains Kibana for visualization and Elasticsearch for consolidation of the monitoring metrics. Through the capabilities of Elasticsearch one can horizontally scale and clustermultiple monitoring components. The client container contains the actual deployment of the tools that have to bemonitored. In this particular instance it contains Logstash,Hadoop and the collectd module. The Logstash connects toElasticsearch cluster as client and stores the processed andtransformed metrics data there.

Basically the proposed solution consists of a collection oftools that are used in order to montior different metricsfrom different layers. One of the main advantages of thissolution is the ease of adding and removing different components from the system. Another interesting aspect of thisarchitecture is the ease with which one can extract differentinformations from tools.

Hadoop Vaidya [8] (Vaidya in Sanskrit language means "onewho knows", or "a physician") is a rule based performancediagnostic tool for MapReduce

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

jobs. The mechanism behindVaidya is to perform post analysis steps for map-reduce jobs.For this purpose it parses and collects different executionstatistics from job history and different configuration files.

Ganglia [5], is a scalable distributed monitoring system forhigh-performance computing systems such as clusters andGrids. The main target for Ganglia is federation formationof clusters and is based on a hierarchical design. Gangliaheavily 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 toits design it manages to achieve very low per-node overheadas well as high concurrency.
The tool is designed such that it is robust and easy to port todifferent operating sustems. While developed it was portedto a vast set of poerating systems and processor architecture.Currently it is in use for thousands of clusters around theworld. One of its major pluses it the cappability to scaleup in order to handle clusters that consists of thousandsof nodes.Currently being used to connect clusters fromdifferent university campuses around the world.

The Apache Ambari [1], is yet another tool that aims atmaking Hadoop management simpler. Ambari project isdeveloping software that for different tasks as provisioning,managing and monitoring Apache Hadoop clusters. At itscore, it also provides an easy to use and intuitive Hadoopmanagement web user interface trough RESTful APIs.

Apache Chukwa [2] is an open source data collection systemfor monitoring large distributed systems. Chukwa is builton top of the Hadoop Distributed File System (HDFS) andMap/Reduce framework. Since it uses these technologies itis easily scalabel and robust. Besides collecting the monitoring data, it also provides a powerfull toolkit that allow usersto monitor, display and analyze results of different runs inorder to better understand the collected data. The tool isreleased under Apache 2:0 licence.

Datastax [3] provide a solution, OpsCenter [16], that can beintegrated in order to monitor Cassandra installation. Using OpsCenter on can monitor different parameters of theCassandra instance and also different parameters providedby the actual machines on which it runs. Also, OpsCenter exposes an interactive web UI that allow administratorsto add/remove nodes from the deployment. An interestingfeature provided by the OpsCenter is the automatic loadbalancing. For integration of OpsCenter with other toolsand services an developer API is provided.

B. Monitoring requirements

In the DICE project there is a need for a monitoring anddata warehousing solution. It must be able to collect andserve monitoring data from a variety of big data frameworks.Its main function is to aggregate data and to serve it to avariety of different DICE tools. In essence the monitoringsolution has to be able to collect monitoring data acrossmultiple cloud layers.notherusecase thatrequires a certain degree of autonomous behaviour is whenan 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 ofthe underlying hardware (or VM whichever the case). Thiscan be done using a software agent which once uploaded intoa target VM (i.e. via SSH). This software is able to detectall 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 andavailable 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 duringthe design and implementation of the DICE monitoring anddata-warehousing solution.

REFERENCES

1] Apache ambari. https://ambari.apache.org.

[2] Apache chukwa. chukwa.apache.org.

[3] Datastax. http://www.datastax.com.

[4] Elasticsearch. https://www.elastic.co.

[5] Ganglia. http://ganglia.info.

[6] Hadoop monitoring.http://blog.sequenceiq.com/blog/2014/10/07/hadoopmonitoring/.

[7] Hadoop toolkit. https://code.google.com/p/hadooptoolkit/wiki/HadoopPerformanceMonitoring.

[8] Hadoopvaidya.http://hadoop.apache.org/docs/r1.2.1/vaidya.html.

[9] Kibana. https://www.elastic.com/producs/kibana.

[10] Logstash. http://logstash.net.

[11] Manage engine. https://www.manageengine.com/.

[12] Mms mongodb. https://mms.mongodb.com/.

[13] Nagios. https://www.nagios.org/.

[14] Open services for lifecycle collaboration.http://open-services.net.

[15] Opennebula. http://opennebula.org/.

[16] Opscenter. http://www.datastax.com/what-weoffer/products-services/datastax-opscenter.

[17] Sequenceiq. http://sequenceiq.com/.

[18] Server density.https://www.serverdensity.com/plugins/mongodb.

[19] G. Aceto, A. Botta, W. De Donato, and A. Pescap`e.Survey cloud monitoring: A survey. Comput.Netw.,57(9):2093{2115, June 2013.

[20] K. Alhamazani, R. Ranjan, K. Mitra, F. Rabhi, P. P.Jayaraman, S. U. Khan, A. Guabtni, andV. Bhatnagar. An overview of the commercial cloudmonitoring tools: Research dimensions, design issues,and state-of-the-art. Computing, 97(4):357{377, Apr.2015.