IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

A SOLUTION FOR THE DEVELOPMENT OF DISTRIBUTED APPLICATIONS AIMEDAT AUTOMATIC RESOURCE MANAGEMENTIN THE CLOUD COMPUTING SCENARIO

Jaideep Bishnoi[1], Shivani Rana[2], Prof. Dhirendra Sharma[3]

[1]*Research Scholar, MTech (CES), Himachal Pradesh Technical University, Hamirpur*
[2]*Lecturer, Department of Computer Science & Engineering, Himachal Pradesh Technical University, Hamirpur*
[3]*Professor, University Institute of Technology, Himachal Pradesh University, Summerhill, Shimla*

Abstract
Cloud Computing has come of age later in 2006, Amazon was the first to market with cloud services of its kind. There are hints that subscription to cloud services by the local enterprises will soon be on a skyrocket track, notwithstanding a modest start in beginning years. With its wide-ranging implications in many areas of computing, notably big data, cloud computing is currently at the top of any list of subjects in computer science. Cloud computing, the life time ambition of computing as a utility, has the power to convert a big part of the IT sector, making software even more attractive as a service and altering the way IT gear is built and purchased. Developers with new invention ideas for new Internet services no longer require the significant capital outlays in hardware to implement their service or the human expense to operate it. They should not be afraid about overprovisioning for a service whose popularity does not meet their forecasts, thus squandering costly resources, or under provisioning for one that becomes incredibly popular, thus missing potential consumers and money. Moreover, businesses with massive batch-oriented operations can achieve results as quickly as their programs can scale, as the cost of running a thousand servers for an hour is the same as running a single server for a thousand hours. The IT industry has never seen something like this before in terms of scalability without incurring additional costs.

INTRODUCTION

Cloud computing is a concept that has been widely disseminated and discussed over the past few years. Although his first known academic record dates back to 1997 (CHELLAPPA, 1997), only in 2007 the term began to appear in historical search records in one of the most popular search engines on the Internet, Google. In the business environment, this is one of the most prominent and relevant topics among company managers, occupying the top positions in the research rankings and business strategies in the last four years (MCDONALD, 2013).

Despite being a popular topic, there remains confusion about what cloud computing really is and what its usefulness is. Generally speaking, the term refers to a new computing model in which software and hardware are made available in the formof services. (ARMBRUST et al, 2010) makes clear distinction between two types of services. The first, called utility computing, is defined as a set of hardware and software services related to the operation and management of cloud resources (processing, storage, and networking). The second type, areapplications provided as a servicethrough the Internet.

Among the innovative characteristics of cloud computing, (ARMBRUST et al, 2010) cites the dynamic acquisition and release of computational resources according to the needs of users (on demand), quickly and without the need for a prior announcementbythe user of the services.

TRADE WIND SOLUTION

the study and analysis of the Wind application, characteristics of the solution were observed that could be generalized to elasticity management systems through a new approach. Apart from the generic functionalities of the specific ones, analysis is allied to the survey of services common to elasticity management solutions, it was envisioned the possibility of developing a middleware and the availability of an API for the aces resource management and elasticity services. Such a solution could be expanded with the insertion of new services.

Literature Review

Dan C.Marinescu (2018) Multi-objective optimization in the cloud places a premium on careful policymaking and decision-making. Due to the massive nature of cloud infrastructure and the unpredictability of system interactions with a broad user base, efficient resource management presents a significant challenge. It is impossible to have precise knowledge of the system's global status, and because to the sheer number of users, it is extremely difficult to foresee the nature and extent of the system's burden.

GALANTE; BONA, (2012) From the solutions that fall into the categories with the highest number of related papers, according to the work of, the studies aimed at automating the mechanisms of elasticity were selected, prioritizing solutions to Bert as and academics, for which it is possible to study the solutions in detail. From this first selection criterion it is possible to draw an overview of infrastructure solutions (IaaS). Of the 18 solutions that are in the infrastructure category, 6 are predictive automatic policy solutions and 2 manuals. Of the 10 remaining solutions, 5 are proprietary and one of the academic papers was not available free of charge. Analysing the last 4works, it was observed that one of them was classified as an infrastructure solution, when in reality it is a platform solution. The remaining solutions are reviewed below.

(Buyya and Gill, 2018)Commercial cloud providers such as Microsoft, Google, and Amazon heavily depend on datacentres to support the ever-increasing demand for computational requirements of their services. Such demand has subsequently increased operational costs of running large infrastructures, as well as producing substantial carbon emissions that negatively impact the environmental sustainability of cloud services

CHAPMAN et al, (2010) proposes solution called RESERVOIR (acronym for Resources and Services Virtualization without Barriers). This solution proposes an architecture where the service/application is defined as a set of software components, encapsulated in the form of one or more virtual machine images (O, middleware, applications, configurations, and data). RESERVOIR also provides a syntax and framework for defining static rules of elasticity, ensuring their dynamic management.

FITÓ et al, (2010) The solutionpresents system of elasticity that aims at compliance with SLAs and considers economic variables for decision making. Unlike the metrics commonly monitored in infrastructure elasticity systems that use processor utilization percentage measurements or the number of reads and writes to disk, its main metric is the response time of requests, and the measurement is repeated in an interval of 10 seconds.

KRANAS et al, (2012)proposes a generic framework for a managing elasticity in any environment or application for cloud computing, introducing the concept of elasticity as a service. The work describes an architecture and its components, but unfortunately does not implement any prototype to validate the proposal, and does not propose solutions of how to describe the rules for the increased or decreased escalation.

MARSHALL et al, (2010)presents an interesting solution for elasticity of Web applications that use the client-server model. This is done by adapting traditional services such as scheduling data batch processing, data storage, and web services. Its main elasticity mechanism is key to monitoring a queue of tasks to be performed by the resource cluster. Soevery time as the task queue begins to grow, a resource manager adds new processing instances to the cluster.

Motivation

WhilePaaS solutions are efficient and facilitate the work of developersfor developing applications adapted to the cloud computing environment, each proprietary solution has its listofservices andAPIs with different forms of use, implying some dependence bypart of the user (GALANTE; BONA, 2012). Another factor to be considered in these platforms is the difficulty in maintaining the flexibility of configuration of applications and their structures and, at the sametime, providing characteristics inherent to cloud computing such as scalability and fault tolerance (ARMBRUST et al, 2010).

In addition, some application domains, such as distributed systems, have characteristics that depend on control and interaction with infrastructure resource management modules to perform better. In the case of content-provided applications, network resources are limited and must be managed to avoid waste. This type of application uses data processing and storage capabilities to reduce the utilization of network resources. Such applications are commonly implemented incomputingclouds, wheredataprocessingand storage resourcesare mostwidelyusedand are close (in oneprocessing centre, same server rack).

Objectives

This work aims to develop a solution that helps compute cloud providers improve the management and operation of their distributed application support infrastructure. More precisely, the goal is to specify, design, and evaluate an architecture in which serviceprovision considers prioritizing resource utilization according to your needs. For this goal to be achieved, the following specific objectives are opposed:

- To raise with the literature the infrastructure and platform solutions forcloud computing that perform some kind of dynamic management of resources allocated to the applications being executed;
- Study thecharacteristics of applications for cloud computing that allow the elasticity of resources;
- Specify functionality relevant to a cloud computing platform that assists in managing the use of cloud resources elasticity;
- Design a solution by grouping these features;
- Develop a prototype that allows the validation of the proposed solution.

Method

This work used the applied research method based on hypothesis-deduction, using scientific references for problem definition, specification of the solution hypothesis and its evaluation. In addition to the bibliographic survey of existing solutions and the specification of functionalities for the proposed solution, a prototype was implemented to validate the hypothesis that "it is possible to dynamically optimize the use of cloud resources by distributed applications using a management solution that takes into account consideration the specifics of the target application".

In order to raise and evaluate the proposals related to the problem domain, open cloud computing solutions and their characteristics were studied. Such studies allowed the identification of concepts related to the investigated areas and the main challenges to be addressed in the solution specification. In particular, functionalities common to several elasticity management applications were identified, leading to the definition of a reusable module for the elasticity of resources in different contexts. From these studies, specific applications were chosen to provide real-time video streams. The interest in this particular class of applications is justified not only by the high data flow involved in them, but also by the fact that there are several studies to optimize their resources, such as multicast mechanisms implemented in the application layer. Using this application as a case study, a solution was then adapted for automatic reduction of redundant flows with specific functionalities in this context. In addition to the proposal of an elastic application development model, an application composition model and a middleware, a prototype was implemented based on the requirements raised in the problem specification. Finally, performance data were collected in order to demonstrate the correct functioning of the proposed solution.

Architecture

The solution is organized into an architecture composed of five different application components: Tail Node, Intermediary Node, Head Node, Statistics Server, and Visualizer. Figure illustrates theorganization of the architecture based on the three nodes described below:

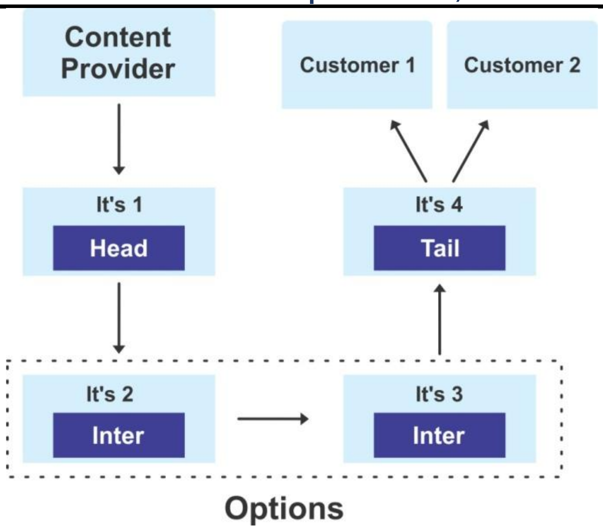

Figure 1: Interaction betweenwindarchitecture nodesfordistributing avideo stream

- **Tail Node is** the node responsible for receiving flow requests from clients, forwarding them to a Tail Node or Intermediary Node. When thisnodereceives a video stream, it forwards it to the requesting clients;

- **Intermediary Node is**a type of node that receives flow requests from tail nodes or even Intermediary Nodes and forwards them to Head Node or Intermediary Node nodes. When you receive a video stream, this node forwards it to Intermediate Node or Tail Node nodes;

- **Head Node is** a type of node that receives flow requests from tail nodes or Intermediary Nodes and requests a video stream from the content provider. When you receive a stream of content from the content provider, this nodeforwards thedata tonodes of the Intermediary Node or Tail Node type. In addition to the node responsible for handling requeststhereare two types of nodesthat fulfill important roles in architecture: the following Statics Serverand Visualizer.

- **Statistics Server is** a server responsible for keeping up-to-date information about thecurrent state of theapplication, indicating whichtopology nodes have active instances;

- **Visualizer is** a graphical interface that allows you to monitor, in real time, the trafficgenerated by the application in the network topology.

In addition toforwarding requests andstreams, intermediary Node and Head Node nodes are also responsible for verifying the amount of requests tothe same video stream;if there is more than one request, themechanism checks the possibility of reducing redundant flows and instantiates anew Intermediary Node application node in a position of topology that optimizes flowdistribution using the longest common path algorithm. As parameters of this algorithm, the topology configured through a specified file and the current state of the network topology formed by those of the active instances are indicated.

EXPERIMENTAL ANALYSIS

Testing Method

Two generating sources were used to collect the analyzed data. The first source was the application's event log files, automatically generated from the implementation of additional modules present in middleware and the video distribution application itself. The second source was the data collected by the TCP Dump application, which, when executed, generates records of all traffic sent and received by the interfaces of a monitored machine. Because one of the metrics specified for application validation is time-based, and application nodes are distributed across different hosts and virtual machines, it was necessary to perform the process of synchronizing the clocks of these computers. The protocol used for this synchronization was the Networking Time Protocol (NTP). After synchronization of the clocks, it was possible to analyse the behaviour of the solution from the event log in chronological order, even on different machines.

The compilation of network traffic data obtained by TCP dump was performed automatically, through the creation of filters in the Wireshark tool. Such filters allow only the data of interest for analysis to be made available after processing the original log fileTo ensure a minimum level of consistency of the collected data, each experiment was repeated 10 times. To perform the repetitive tasks of execution of the experiment, collection and analysis of the data, a script was implemented in the Shell language, native in environments that use the operating system Linux, linux.

Results of Scenario 1

The following are the results obtained on: the time required to reduce redundant flows, from detection to completion of optimization and the network band used during the experiment.

Redundant Flow Reduction Time

From the data collected by the application's event log files, the necessary times were observed for: the detection of a redundant request (RR), the collection of information about resource topologies next to the Core module, the calculation of the optimization point by the LCP Module, the creation of the virtual machine (MV), the startup of the application operating system (boot), and the time for redirecting the flows to the newly instantiated node. Table 1 presents each of the events and their duration time, that is, the time required to complete the task and the time the event was recorded during the test run.

Table 1: Results for the time metric for reducing redundant flows

Event	Duration (s)	Time (s)
Start of the test		0
RR detected	10, 004	20, 004
Information	0, 155	20, 159
LCP	0, 012	20, 171
MV Creation	74, 494	94, 665
Boot	45, 004	139, 741
Redirect 1	0, 523	140, 264
Redirect 2	0, 044	140, 308
TOTAL	130, 236	

As can be seen inTable 1, the tasks of obtaining the information (~10 seconds), the creation of the virtual machine (~74. 5 seconds), and the startup of the Operating System and applications (~45 seconds) are the most significant times in the composition of the reduction time redundant flows to scenario 1. The remaining measurements were all below the order of seconds. Figure 2 shows the composition of the time of reduction of redundant flows by the most significant tasks.

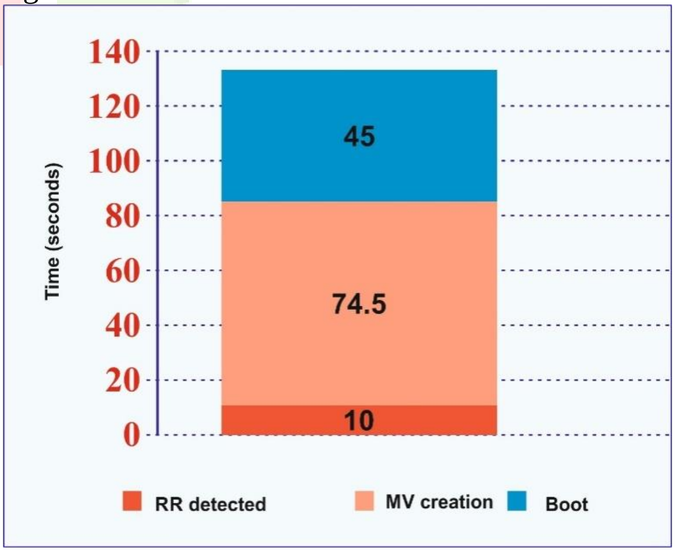

Figure 2: Composition of the time of reduction of redundant flows

In the study of (MAO; HUMPHREY, 2012), the average startup times of virtual machines in major infrastructure providers were raised. The results show that the values vary approximately from 44 to 810 seconds. From Figure2, it is possible to observe that of the 130 seconds (2 secondsminutes and 10 seconds) required to reduce redundant flows, about 120 seconds are due to the creation and startup of the virtual machine (within the expected values) and only 10 seconds for the application to detect the existence of redundant requests. Thus, the time cost of the Trade Wind solution represents only 8% of the total time of reducing redundant flows.

Network Bandwidth Economy
For the evaluation of bandwidth consumption metrics, the following applications were used: TCP Dump, for traffic data collection and Wireshark, for filtering and analysis of the files generated by TCP Dump. The band consumption data were collected in twoand two topology places between routers B and C (representing total traffic); and between router C and vHost 3 (representing Client 1). Figure 3 shows the first 200 seconds of the video stream delivery (duration of ~598 seconds).

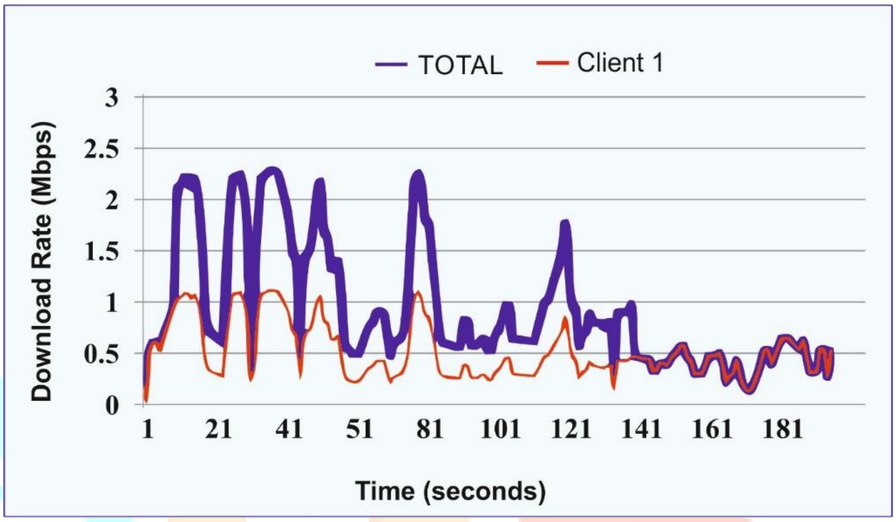

Figure 3: Data traffic analysis in scenario 1

From Figure 3, you can observe the moments when the events of the description of scenario 1are triggered. The test begins with the acquisition of the first video stream by Client 1 and the request made by Customer 2 after 10 seconds. Note that total traffic doubles (total at 2. 5 MBps and Client 1 at 1. 25 MBps) and lines track client link traffic 1. Approximately 130 seconds later, the flows are reduced (close to instant 141) and the total rate Linh meets the Customer Rate Line 1. As can be seen in the results presented, resource savings are directly related to the number of requests sent from different Distribution nodes. The greater the number of distribution nodes required and the duration of the acquisition of video streams of the video, the greater the potential for bandwidth savings in the central topology links by reducing the redundant streams transmitted.

Results Scenario 2
The protocol used to provide video streams was HTTP, in which there is no direct control over data transmission rate. This control is performed by TCP protocol flow control mechanisms, such as the variation of the size of the transmission window. Scenario 2 was defined with the objective of evaluating the behavior of the Trade Wind solution for the acquisition of video stream through the use of different content acquisition applications.

The standard application used in the tests, VLC Media Player, performs the acquisition of the stream according to the bitrate required for the video viewing (bitrate). The WGET application, developed for downloading data from a command-line terminal, aims to obtain the data as quickly as possible. The data for evaluation of the total time and download rate of the content were measured from the link between routers B and C of the topology. As in the tests performed in scenario 1, the TCP Dump and Wireshark applications were used for data collection and analysis. Figure 4 illustrates the download rate of video content using the VLC player and WGET applications between instant 0 and 201. The separately measured results were grouped in the same chart for comparison between the different download rates and different download times.

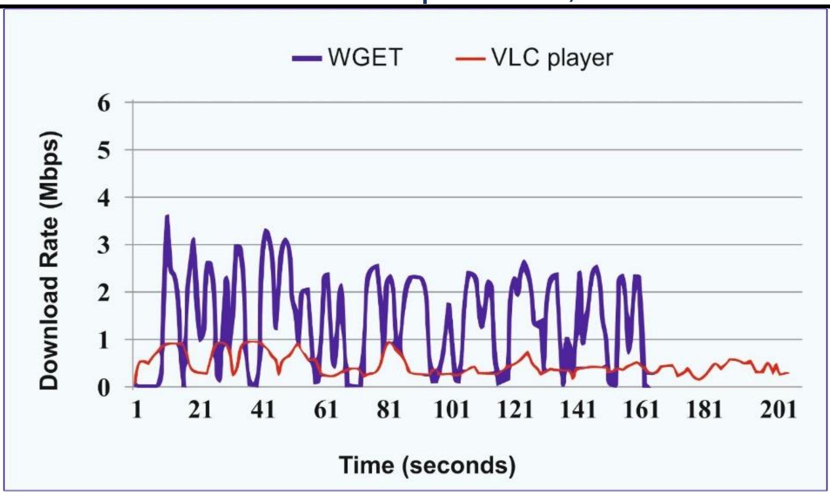

Figure 4: Data Traffic Analysis in Scenario 2

As can be seen in Figure 4, the download rate practiced by the WGET application is much higher than the rate practiced by the VLC application. On average, the rate measured in execution with WGET is ~1. 9 MBps, while with VLC it is ~0. 6 MBps, confirming the characteristics described at the beginning of this subsection. The VLC has a download rate technically equal to the average video transcoding rate (0. 615BMps) indicated in Table 14. Due to the difference in download rates, the total download times were also different. While theWGET append-up ended the content download in ~153 seconds, the VLC used ~596 seconds (two seconds less than the video duration time).

Based on the observed results, it can be stated that there is a need to implement a temporary block storage mechanism if the application for redundant trade wind flow reduction is used in conjunction with a flow delivery service video under demand. Such adaptation is necessary due to the different settings of download rates found in http video stream viewing applications. In the case of the test presented in scenario 3, the client and using VLC Media Player as a content viewer downloaded the last blocks of content about 440 seconds (7 minutes and 33 seconds) after the WGET application. In non-accurate calculations, considering the 350MB video size and the 10-minute duration time, approximately 35MB of content per minute is transferred. The temporary block storage engine would have to support a value of 256. 67MB of data on each Relay node of the video stream distribution application. Another possible adaptation would be limiting the download rate to a value that would help decrease the size of the list of stored temporary blocks (e. g., limit the download rate in a video video video).

Scenario 3 - Multi-channel

Once the elastic application development model was proposed and the functionalities that should be part of the middleware of the Trade Wind solution were identified, an information management mechanism was implemented about available and allocated resources, users and their instances of application, and configuration information about the applications developed. In particular, theinformations on the resources allocated in the form of virtual machines and application instances, allowed the addition of a new functionality to the application of supply of video streams in real time with reduction of redundant flows developed in the Solution Wind: the creation of multichannel. In other words, with the management of this information it was possible to provide independent and simultaneous video streams, even application instances using common hosts. The tests of this subsection are intended to validate the functionality of multichannel

As in scenarios 2 and 3, the TCP Dump and Wireshark applications were used to collect and analyze the data. Data collection was performed by monitoring the link between routers B and C, from the topology presented in Subsection 5. 3. 2. Figure 5polishes the variation of the download rate measured for video stream 1 (line in blue), for video stream 2 (line in red) and total flow (in yellow), between instants 0 and 101.

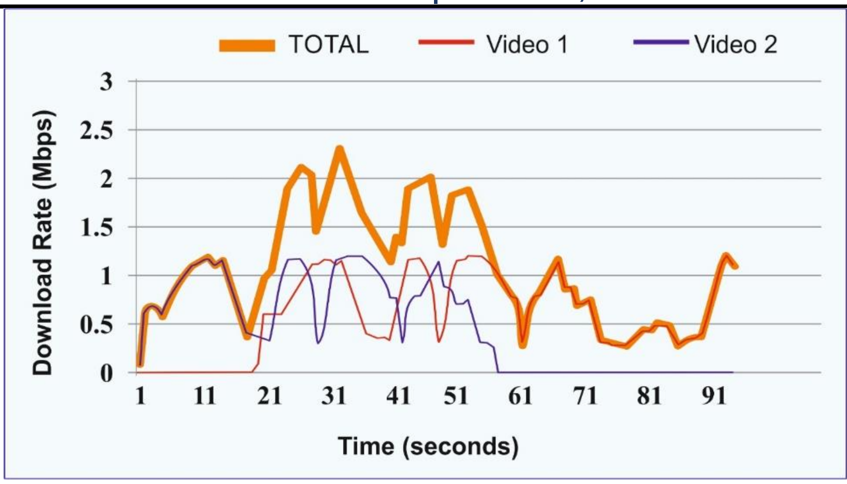

Figure 5: Data traffic analysis in scenario 3

From Figure 5, you can observe the moments when the events described in scenario3are triggered. The test begins with the acquisition of video stream I by Client 1. The acquisition request of video stream II is performed by Customer 2 after 20 seconds. Note that total traffic increases at the same time, and the shape of the curve does not resemble any of the two curves. Traffic decreases again at instant 60, with the transmission of simultaneous flows to different channels.

Solution Limitations Analysis

The proposed modifications in the Trade Wind solution to allow the separation of functionalities common to elastic applications in the organization of a middleware that offers them in the form of service, introduced additional elements in the application of supply of video streams in real time with the reduction of streams impacting cloud resource consumption. This subsection is dedicated to discussing the impacts of trade wind design decisions and their limitations.

Limitation of Video Stream Type

Through the results obtained in the tests performed in scenario 3, it was observed that when requesting customers use different applications for the acquisition of the same video stream, the variation in the standard values for download rates makes it impossible to transmit streams under with the current organization of the application of redundant flow reduction. For the transmission of video-on-demand streams, it is necessary to implement temporary block storage mechanisms on all nodes of themechanisms for restricting the download rate of devices with higher rates.

Conclusion

This work presented Trade Wind, a solution that allows the development of distributed applications and services for the automatic management of resources and elasticity in compute clouds. Through a study on elasticity solutions for computing clouds, it was observed the concentration of elasticity solutions focused on the management of processing and storage resources for applications of the client-server type. These solutions implement self-attend resource management through infrastructure mechanisms, through automatic reactive policies.

Automatic reactive policies use rule description to define metrics to monitor, and that reached a borderline value, trigger actions to adjust cloud resources to new demand. To ensure the functioning of the rules, two mechanisms are recurrent in the description of elasticity solutions: the mechanisms of definition and monitoring of metrics and the mechanisms of elasticity control. Commonly implemented in the cloud infrastructure layer, these solutions encounter certain difficulties in defining and monitoring rules to meet variable demands.

Based on these observations, an elastic application development model was based on four functionalities: definition and monitoring of metrics and rules conditions, verification of the possibility of optimization, management of information about resources and applications and the execution of elasticity actions. From this model, we defined the specific functionalities, which should be implemented by developers and the functionalities common to elastic applications, which could be reused in the form of a middleware.

Starting from the definition of middleware and its services, a complete solution of resource management and elasticity called Trade Wind was proposed, which can be extended according to a application composition model from the implementation of features and services. As proof of concept, the application of providing video streams with automatic reduction of redundant streams for operation from middleware was adapted. For the evaluation and validation of the proposed solution, a test environment composed of execution scenarios, a topology, a set of metrics to be evaluated, the characterization of the video content used and the detailing of the method of performing the tests were specified and implemented.

The results obtained validated the function of the application of proof of concept adapted for the operation in conjunction with Trade Wind, as well as its additional functionality of providing video streams in multichannels. The results also indicated that Trade Wind and the application d and reduction of redundant video streams impact only 8% of the total time of adaptation of the infrastructure, being the time of creation and startup of the virtual machine responsible for the longest time consumed in this process. The application of redundant flow reduction s also proved to halve the bandwidth consumption in the test scenario, having the potential for greater savings in the case of adding more redundant flows.

Future Work

Due to the time limitation and the need for priority/focus of the study, some results previously related to this research work had to be waived. In this section, such related studies are presented and discussed superficially.

- Measurement of solution impact values on the performance of the real-time video stream delivery application: processing overhead, memory overhead, and response time of the real-time video stream delivery service for the various scenarios proposed;
- Refinement of the hardware and software requirements of virtual machines responsible for host resource control/management mechanisms (virtual machines running the Core module): by creating virtual machines with different configurations, proposing a minimum amount of resources needed to implement the solution;
- Assessment of the impact of the decrease in available resources on system response time (reaching the optimal amount of resources to be used)
- Test execution in an environment closer to the actual (no virtualization of resources such as rotators and switches);
- Extension of platform-level services for various applications, for example online games, load balancing, among others;
- Creation of access control model to mediate API and Middleware.

References

[1]. Dan C. Marinescu,Chapter 9 - Cloud Resource Management and Scheduling,Editor(s): Dan C. Marinescu,Cloud Computing (Second Edition),Morgan Kaufmann,2018,Pages 321-363,ISBN 9780128128107

[2]. Gill, S. S., Garraghan, P., Stankovski, V., Casale, G., Thulasiram, R. K., Ghosh, S. K., ... &Buyya, R. (2019). Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *Journal of Systems and Software*, *155*, 104-129.

[3]. GALANTE, G.; BONA, L. C. E. A Survey on Cloud Computing Elasticity. In: Proceedings of the 2012 IEEE Fifth International Conference on Utility and Cloud Computing, Chicago, IL, USA: IEEE Computer Society, [S.I.], 2012. (UCC) p. 263- 270. ISBN 978-1-4673-4432-6. Disponívelem: <http://dx.doi.org/10.1109/UCC.2012.30>|

[4]. FITÓ, J. O.; GOIRI, I.; GUITART, J. SLA-driven Elastic Cloud Hosting Provider. In: Proceedings of the 2010 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, Pisa, Italy: IEEE Computer Society. [S.L], 2010. (PDP). p. 111-118. ISSN 1066-6192. Disponívelem: <http://dx.doi.org/10.1109/PDP.2010.16>.

[5]. KRANAS, P. et al. Elaas: An innovative Elasticity as a Service framework for dynamic management across the cloud stack layers. In: Proceedings of the 2012 Sixth Intemational Conference on Complex, Intelligent and Software Intensive Systems, Palermo, Italy: IEEE Computer Society, [S.I.] 2012. (CISIS). p. 1042-1049. ISBN 978- 1-4673-1233-2. Disponívelem: <http://dx.doi.org/10.1109/CISIS.2012.117>.

[6]. MARSHALL, P.; KEAHEY, K.; FREEMAN, T. Elastic Site: Using Clouds to Elastically Extend Site Resources. In: Proceedings of the 2010 10th IEEE/ACM Intemational Conference on Cluster, Cloud and Grid Computing, Melbourne, Australia: IEEE Computer Society, [S.I.], 2010. (CCGrid). p. 43-52. Disponívelem:

<http://dx.doi.org/10.1109/CCGRID 2010.80>.
[7]. S.S. Gill et al. ROUTER: Fog Enabled Cloud based Intelligent Resource Management Approach for Smart Home IoT Devices
[8]. J. Syst. Softw. (2019) S. Taherizadeh et al.Monitoring self-adaptive applications within edge computing frameworks: a state-of-the-art review
[9]. J. Syst. Softw. (2018) S. Singh et al. Q-aware: quality of service based cloud resource provisioning
[10]. CHENG, B. MediaPaaS: a Cloud-based Media Processing Platform for Elastic Live Broadcasting. In: Proceedings of the 2014 7th IEEE International Conference on Cloud Computing, Anchorage, AL, USA: IEEE Computer Society. [SI], 2014. (CLOUD). p. 713-720. Disponivelem: http://dx.doi.org/10.1109/CLOUD 2014 100
[11]. JebliImane, Belouadha Fatima-Zahra, Kabbaj Mohammed Issam and Tilioua Amine, "Prediction of solar energy guided by pearson correlation using machine learning[J]", *Energy*, vol. 224, 2021.
[12]. X Ji, X M Gong and Y H Dai, "Video Emotion Classification Algorithm Based on Brain Reponse[J]", *Computing technology and automation*, vol. 37, no. 02, pp. 63-66, 2018
[13]. P F Huang, H D Wang and Y C Jin, "Offline data-driven evolutionary optimization based on tri-training[J]", *Swarm and Evolutionary Computation*, vol. 60, 2021.
[14]. Pandey, S., Voorsluys, W., Niu, S., Khandoker, A., &Buyya, R. (2012). An autonomic cloud environment for hosting ECG data analysis services. Future Generation Computer Systems, 28(1), 147–154. https://doi.org/10.1016/j.future.2011.04.022
[15]. Sultan, N. (2013). Cloud computing: A democratizing force? International Journal of Information Management, 33(5), 810–815. https://doi.org/10.1016/j.ijinfomgt.2013.05.010
[16]. Sultan, N. (2014). Making use of cloud computing for healthcare provision: Opportunities and challenges. International Journal of Information Management, 34(2), 177–184. https://doi.org/10.1016/j.ijinfomgt.2013.12.011
[17]. Svantesson, D. J. B. (2012). Data protection in cloud computing – The Swedish perspective. Computer Law & Security Review, 28(4), 476–480. https://doi.org/10.1016/j.clsr.2012.05.005
[18]. Thilakanathan, D., Chen, S., Nepal, S., Calvo, R., &Alem, L. (2014). A platform for secure monitoring and sharing of generic health data in the Cloud. Future Generation Computer Systems, 35, 102–113. https://doi.org/10.1016/j.future.2013.09.011
[19]. Varadi, S., Kertesz, A., & Parkin, M. (2012). The necessity of legally compliant data management in European cloud architectures. Computer Law and Security Review, 28(5), 577–586. https://doi.org/10.1016/j.clsr.2012.05.006
[20]. Venters, W., & Whitley, E. A. (2012). A critical review of cloud computing: researching desires and realities. Journal of Information Technology, 27(3), 179–197. Journal Article.