



An Elastic and Scalable Data Streaming System for Distributed Cloud

S.B. Daniel Barnabas M.E.,¹

Assistant Professor, Department of Computer Science and Engineering, CSI College of Engineering, Ketti - 643 215, India¹

ABSTRACT: In cloud computing environment, dynamic resource allocation and reallocation are keys for accommodating unpredictable demands. Online processing of continuous data flow produce very high load. Current system does not scale with input load due to single-node bottlenecks. This happens because of static configuration leading to under or over utilization. This article explains about how bottlenecks could be avoided in a distributed cloud, where application developers can selectively lease distributed resources. This could be handled by enabling effective adjustment of resources to the incoming load which is by facilitating cloud through dynamic load balancing and elastifying.

KEYWORDS: Data streaming, scalability, elasticity, Decommissioning

I. INTRODUCTION

Current cloud computing providers mainly rely on large and consolidated datacenters in order to offer their services. This centralized infrastructure brings many challenges like need for resource over provisioning and costly heat dissipation and temperature control and also naturally increases average distance to end user [11]

In the last few years, there have been substantial advancements in the field of data stream processing. From centralized Stream Processing Engines (SPE) [6], the state of the art has advancement to SPEs able to distribute different queries among a cluster of nodes or even distributing different operators of a query across different nodes [4]. However, some applications have reached the limits of current distributed data streaming infrastructure. Many applications exhibit sudden change in the workload that can result in a variation of order of magnitude between peak and valley loads.

A parallel processing engine deployed on a fixed number of processing nodes leads to either under provisioning or over provisioning. Under provisioning results in the violation of service level agreement that incur high economic costs eventually leading to unhappy users and raising bad reputation. Similarly over provisioning is not cost effective but resources are not fully utilized.

Under distributed datacenters one may understandably take advantage to improve cost and performance. However, using public infrastructure for communication between datacenter and end user will be far effective. The drawback with this approach is that it transfers traffic control to Internet Service Providers.

The service engine at cloud datacenters should be elastic and adjust amount of resources to current workload. Moreover, elasticity should be combined with dynamic load balancing. Without dynamic load balancing the system would provision new nodes as a result of uneven distribution. Therefore, the saturation of a single node would lead to unnecessary provisioning of new instances. With dynamic load balancing, new nodes are provisioned only when the system as a whole does not have enough capacity to cope with incoming load.

This article lights on challenges for resource allocation in distributed clouds and a scalable, elastic processing engine providing transparent parallelization.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

II. PROBLEM DEFINITION

In a Cloud context, establishing trust should be based on both identities and properties. Establishing a trust model in any system is quickly seen to be a complex problem. The additional components which contribute to a Cloud environment mean that models for establishing trust, where cloud is distributed in addition to their essential offerings, such as scalable services, on demand usage and pay-as-you-go business plans, distributed clouds also take advantage of geo-diversity and also infrastructure complexity, application interdependencies, and the range and size of stakeholders all contribute to the challenge.

To overcome these limitations, we choose a generic and distributed solution that may be used in the context of many types of services. This is a distributed cloud scenario, where cloud providers hire infrastructure on demand and acquire dedicated connectivity and resources from communication providers. It is important to highlight that the infrastructure may range from routers and links to servers and databases. The processing engine operating on logical data streams should be split into multiple physical data sub streams that flow in parallel, thus avoiding single-node bottlenecks. Communication across different nodes is minimized and only performed to guarantee semantic transparency.

III. PROPOSED MODEL

Resource allocation in distributed clouds, focusing four fundamental points

- . Resource modeling
- . Resource offering and Treatment
- . Resource discovery and monitoring
- . Resource selection

When conceiving a distributed cloud, it is natural for its provider to choose the nature of its offering Software, Infrastructure and Platform as a Service.

3.1 Conceptual Architecture:

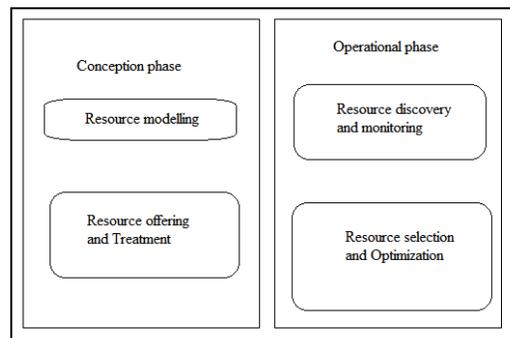


Figure 1. conceptual cloud architecture

Among those four fundamental factors it can be categorized into two phases

- . Conception phase
- . Operational phase

Getting into fundamentals as follows

Resource modeling:

The cloud resource description defines how the cloud deals with infrastructural resources. The granularity of the resource description is important. If resources are described using many details, there is risk that the resource selection and optimization phase could become hard. Adding to it interoperability also stands in path. According to [12], interoperability in the cloud faces vertical and horizontal heterogeneities.

Resource Offering and Treatment:

The RAS must ensure that all requirements may be met with the available resources. These requirements have been defined previously between the provider and each cloud user, and may be represented by service level agreements (SLA) and ensured by the provider through continuous monitoring [15].

Cloud users are able to set inter-node relationship and communication restriction.

Resource Discovery and Monitoring:

Resource monitoring should be continuous and should help with allocation and reallocation decisions as part of overall resource usage optimization.

Monitoring may be passive or active. It is considered passive when there are one or more entities collecting information. If monitoring is active nodes are autonomous and decide when to send asynchronously state to some central entity.

Resource Selection and Optimization:

Resource selection may be done using optimization algorithms. Resource selection strategies fall into a priori and posteriori classes. In the priori case first allocation solution is an optimal one. In posteriori case once an initial allocation there can be a sub optimal solution made.

3.2 Cloud Architecture:

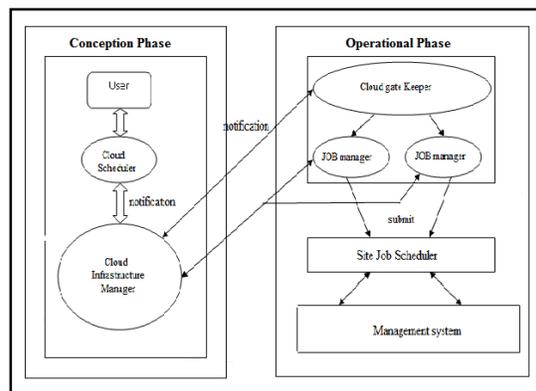


Figure 2. Dynamic cloud architecture

This model performs specialized workload management, job queuing mechanism, scheduling policy and resource management.

The components under Conception phase are as follows,



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

User: One who submits his request.

Cloud Scheduler: This module acts as intermediate between user and cloud infrastructure manager. It accepts the notification from modules beneath and queues job for execution.

Cloud Infrastructure Manager: It is molded with Notification module, Trust management module, organizing DBMS module and Security token service module. This manager assists in providing a platform for doing a job. Though distributed clusters are found Infrastructure Manager provides with dynamic functionality to individual cluster.

The working components under Operational phase are,

Cloud Gate Keeper: Since our scenario is distributed to manage among many clusters we need a gate keeper which checks whether submission is to and from local or foreign cluster.

Job Manager: Currently submitted job(s) is/are organized for scheduled execution. Identifies the job and tracks them according to their notification tag.

Site Job Scheduler: Local scheduling for jobs submitted before it uses the resources are supervised. It schedules the resources available based on free resources.

Management services: Since our environment is distributed it should support scalable and elastic data. In order to provide elasticity we need separate protocol running with new algorithm, provided by new module adding to this infrastructure, which is Elastic manager.

IV. ELASTICITY

Dealing with distributed cloud the working of service provider is so delicate that no flaws should be occurred providing efficient IaaS. In order to support new age of distributed architecture the working standard is to be altered from normal scheduling to peculiar standards. Here we are providing elasticity to our data to move around our infrastructure, which positively influences distributed environment.

The atomic components and its functionality are as follows,

Query: The requests forwarded by user, which is to be processed in the cloud.

Sub Cluster: The distributed environment of cloud providing clusters of resources.

Cloud Instance: Among clusters of cloud which resource is currently been under current utilization for processing data.

Resource Manager: The module which cares about resource and its performance after and before every request.

Elastic Manager (EM): The proposed scenario of computing command is given life by elastic manager, which maintains protocol standard and monitors reaction between resource, data, request and response.

Local Manager (LM): For every cloud instance a manager holds responsible. If a request is triggered then many modules start its execution to take care of load. To serialize these modules Local manager paves path.

Input Manager: Management module that queues input though input might be from same module or from various threads.

Aggregate operator group: Distinguishes threads of request and identifies relation among job request queues.

Filter: Filters related and non related job request. Related request from same module are taken serially.

Map: Module checks capability of resource to execute the given request and submits it.

Load Balancer: Balances load in the resources by checking under or over utilization of resources.

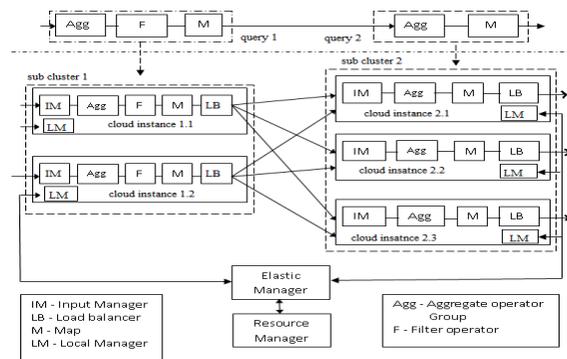


Figure 3. Elastic management architecture

4.1. Elastic Protocol:

Elasticity rules are specified as thresholds that set the conditions that trigger provisioning, decommissioning or load balancing. Provisioning and Decommissioning are triggered if the average CPU utilization is above the Upper Utilization Threshold (UUT) or below the Lower Utilization Threshold (LUT). Reconfiguration actions aim at achieving an average CPU utilization as close as possible to Target Utilization Threshold (TUT).

Load balancing is triggered when the standard deviation of the CPU utilization is above the Upper Imbalance Threshold (UIT).

In order to enforce the elasticity rules, the EM periodically collects monitoring information from all instances on each subcluster.

The information includes average CPU usage (U_i). The EM computes the average CPU usage per sub cluster (U_{av}). If U_{av} is outside the allowed range, the number of instances required to cope with the current load is computed. If the subcluster is under provisioned, new instances are allocated. If the sub cluster is over provisioned, the load of un needed instances is transferred to the rest of instances by offload function.

Load balancing is triggered if $U_{sd} > UIT$ and is based on greedy algorithm.

Elastic Management Protocol:

Upon: new monitoring period has elapsed

- 1: $U_{av} = \sum_{i=1}^n U_i / n$
- 2: **if** $U_{av} \notin [LUT, UUT]$ **then**
- 3: old = n
- 4: n = Compute new configuration (TUT, old, U_{av})
- 5: **if** $n > old$ **then**
- 6: provision (n - old)
- 7: **if** $n < old$ **then**
- 8: free nodes = offload (old - n)
- 9: Decommission (free nodes)
- 10: $U_{sd} = \text{SQRT} \{ \sum_{i=1}^n (U_i - U_{av})^2 \}$
- 11: **if** $U_{sd} > UIT$ **then**



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

12: BalanceLoad (U_{sd})

V. NOTIFICATION MODULE

This module keeps on notifying the cloud scheduler about the job submission to and from cloud infrastructure manager. The Delivery notification is send to user.

Cloud Infrastructure manager (CIM) checks service instances for request submitted by user for existence. If the particular topic exists then CIM uses historic path for successful execution.

The service data set is associated with a service instances. Regarding a service data change, the service data wrapper sends a change notification message to the service data set. If the change is valid for the constraint expressed in the published Topic then a separate thread is created to deliver the change message.

VI. ORGANIZING DBMS

Based upon the principle of keeping the existing data management systems and its interfaces intact, this model attempts to expose the underlying data model and native language API for resource manager.

External Data Resource Manager (EDRM) and Data Resource Manager (DRM) is a relational database management system. The DRM represents the EDRM and binds it to the existing EDRM. It provides start and stop query capabilities.

External Data Resource (EDR) and Data Resource (DR) is a directory in a file system, it exposes the metadata about the external data resource. It must provide access and update query capabilities.

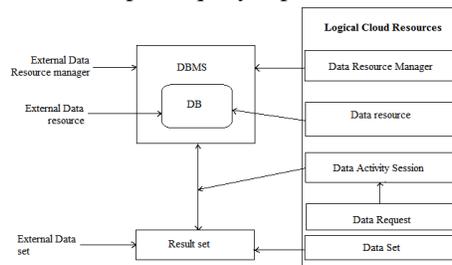


Figure 4. External & logical resources of DBMS

External Data Set (EDS) and Data Set (DS) are the logical database view or a file cache.

Data Activity Session - It is a logical data session fro all data operations.

Data Request - It is logical information regarding a request submitted by requester to data access session. This can be a query, data manipulation activity or others.

VII. SECURE TOKEN SERVICE MODULE

The requester must possess a secure token to establish a secure message channel to the web service end point. Here the service may need to request tokens from other trusted parties called secure token services.

The WS- Trust deals with different aspects of secure token services, including issuance of tokens must be secure and built on top of WS - Security.

WS - Federation defines mechanisms that are used to enable identity, attribute, and authentication and authorization federation across different trust environments.

VIII. EVALUATION

In the set of experiments evaluating effectiveness of provisioning and decommissioning is done.

The LUT = 0.5, UUT = 0.9 and TUT = 0.6 is been set. The load is increased and decreased linearly to observe the effectiveness of provisioning and decommissioning.

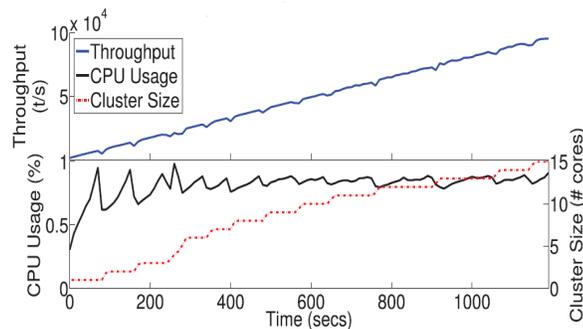


Figure 5. Individual provisioning

This figure Fig. 5. Shows the behavior of the individual provisioning strategy, i.e., when provisioning one instance at a time. The system behavior is studied growing 1-15 instances. The throughput increases linearly with the input load, despite negligible variation at each step. For larger configuration (e.g., 15 nodes), provisioning of one instance results in a negligible increase of the overall computing power, leading to an average CPU utilization close to the upper threshold.

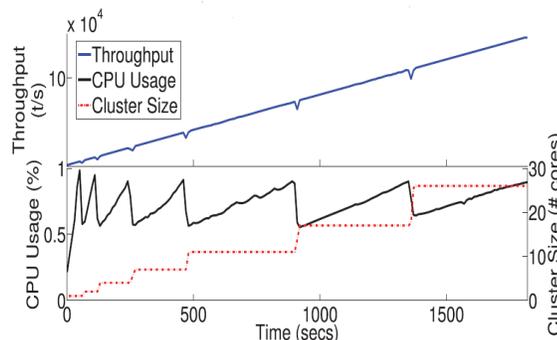


Figure 6. Load aware provisioning

As the number of provisioned node is computed on the current subcluster size and load, each provisioning step achieves the TUT. Fig. 6 Moreover, load-aware provisioning affords less frequent reconfiguration steps than individual provisioning. Hence, the system reaches higher throughput with fewer reconfiguration steps.

Decommissioning of resources from our distributed scalable system also shows good behavior as,

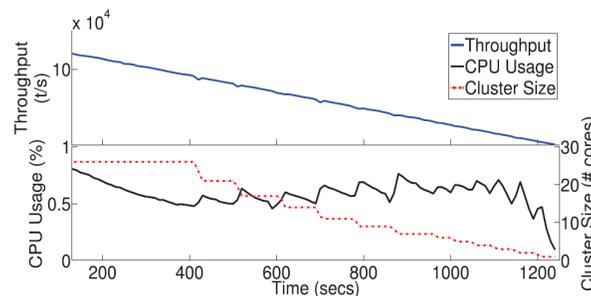


Figure 7. Load aware decommissioning

The decommissioning intrusiveness is lower than provisioning due to decreasing load leading to effective working of system Fig. 6 That is, once instances are transferring ownership the decreasing input rates result in a CPU average utilization slightly below the TUT.

IX. CONCLUSION

The system which is been presented has high scalability and can withstand elastic data flowing through distributed cloud. This system also provides transparent parallelization. Scalability is attained by means of a novel parallelization strategy that minimizes the distribution overhead. Elasticity and dynamic load balancing minimizes number of resources used for copying with varying workloads. The evaluation demonstrates large scalability and effectiveness of a distributed cloud supporting elastic data.

REFERENCES

- [1] Abawajy J. (2011) 'A Dynamic Job Grouping- Based Scheduling For Deploying Applications with Fine-Grained Tasks on Global Clouds' 10th International Symposium on Pervasive Systems, Algorithms and Networks (ISPAN) pp. 784-8
- [2] Abbadi Imad M.(2011) 'Operational trust in clouds' environment. In: MoCS 2011: Proceedings of the Workshop on Management of Cloud Systems. IEEE; June 2011.
- [3] Abbadi Imad M. Toward trustworthy clouds' internet scale critical infrastructure. In: ISPEC in proceedings of the 7th information Security.
- [4] Abadi DJ et al., "The Design of the Borealis Stream Processing Engine," proc. Second Biennial conf. Innovative Data Systems Research (CIDR), pp. 277- 289, 2005
- [5] Bleikertz So`ren, Schunter Matthias, Probst Christian W,Pendarakis Dimitrios, Eriksson Konrad. Security audits of multi-tier virtual infrastructures in public infrastructure clouds. In: Proceedings of the 2010 ACM workshop on Cloud computing security workshop, CCSW '10. New York, NY, USA: ACM; 2010. p. 93e102
- [6] Buyya R., Yeo C. S., and Venugopal S. (2010) 'An Efficient Approach to Task Scheduling In Cloud'In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications Vol 27 pp. 81-94
- [7] Chandrasekaran S et al., "Telegraphcq: Continuous Dataflow Processing for an uncertain World," proc. First Biennial Conf. Innovative Data Systems Research (CIDR), 2003
- [8] Foster I., Kesselman C., Tuecke S. (2010) 'Intelligent workload factoring for a hybrid Cloud Computing Model' International Journal of Supercomputer Applications



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- [9] Gansen Z., Chunming R., Jin L., Feng Z., Yong T.(2010) 'An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers' IEEE Second International Conference on Cloud Computing Technology and Science pp. 97, Nov. 30-2010 - Dec. 3 -2010
- [10] Imad M. Abbadi, Andrew Martin (2011) 'Trust in the Cloud' Department of Computer Science, University of Oxford, United Kingdom information security technical report xxx (2011) pp. 1-8
- [11] Katz Randy H, Konwinski Andrew, et al. Above the Clouds: A Berkeley View of Cloud Computing, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>; 2009
- [12] Sheth A and A. Ranabahu, "Semantic Modeling for cloud computing, Part I," IEEE Computer Society - Semantics & Services, 2010
- [13] Tian L., LING Y. (2010) 'Node Allocation in Cloud Computing Using Optimal Resource Constraint (ORC) Scheduling' International Conference on Computer Application and System Modeling (ICCASM 2010) Vol.7 pp. 567 22-24 Oct. 2010
- [14] Valancius V et al., "Greening the internet with Nano Data Centres," Proc. 5th international conference on Emerging Networking Experiments and Technologies, 2009, pp. 37-48.
- [15] C. A. Yfoulis and A. Gounaris, "Honoring SLAs on Cloud Computing Services: A Control Perspective," Proc. EUCA/IEEE Euro. Control Conf. 2009