

# Elasticity In Cloud Computing

<sup>[1]</sup>Dr.C.Rajabhushanam, <sup>[2]</sup>V.Karthik, <sup>[3]</sup>G.Vivek

<sup>[1]</sup> Professor, Dept of CSE, BIHER, Chennai

<sup>[2]</sup><sup>[3]</sup> UG Student , Dept of CSE, BIHER, Chennai

*Abstract: The biggest challenge of cloud computing is making an application elastic, which consists of variations in load without the involvement of a human manager and without the compulsion to change its code. In this project, we first identified several design issues that have to be addressed, following making an application elastic. Defining the granularity of the elastic components in the application, handling their interconnections and controlling the elasticity at automation era. We have already calculated the contact of autonomous computing and proposed an architecture of an elastic application, considering a perfectly made application surrounded by general concepts and application specific parts. We used a load injection application (Clif) as a use case. Cloud computing is now a completely stable model for providing on demand facilities provisioning to a pay-as-you-go model. Elasticity, one of the major component required for this computing model, is important to handle resources to ensure separate resources to the load variation. However, many works have already measured cloud computing and its features, there is an absence of a comprehensive study of elasticity for the cloud. As an attempt to close this gap, we propose this survey with limitless rational review. We address many aspects of elasticity such as definitions, metrics and tools for measuring, review of the elasticity and existing solutions. This is the first analysis on cloud computing elasticity using a research review.*

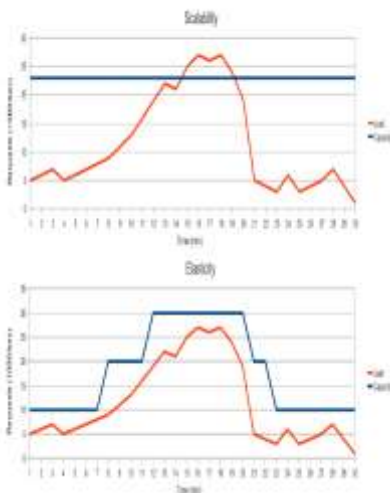
**Keywords:** Elasticity, Cloud Computing, Scalability, Resource Provisioning

## I. INTRODUCTION

The subject of work is to control the scalability of the application deployed in cloud through a particular technique called elasticity. The research domain is elasticity in cloud - systems composed of auto scaling mechanism according to the provided load. The main research is the designing of an elastic virtual instance which can increase or decrease the size considering the data provided. Our main research is to provide a general firm for the paperwork of application elasticity in the cloud environment. We particularly aimed to locate a final service which would:

- Provide responsive ways of auto scaling.
- Minimize the impact of execution of elasticity in variation to strengthen the applications which are programmed

In order to achieve this concept, the plan was to accept an elastic manager, which would be executed as a remove application and make an existing application elastic without modifying its source code. We used a load injector Clif as a use case.

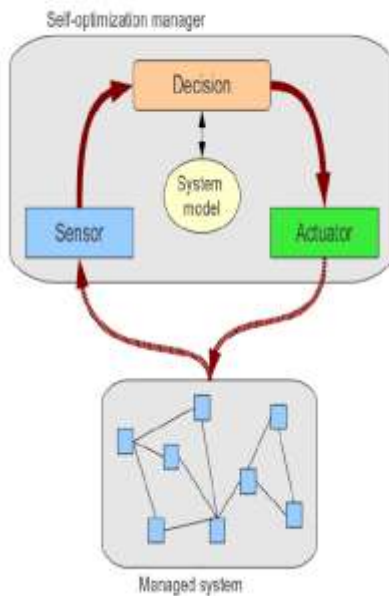


**Figure I.1 – Comparison of classical scalability and elasticity approaches**

## 1) EXPERIMENTAL DETAILS

### APPLICATION ELASTICITY

The variations in load connection is a specific challenge as regarded to the subject of applications in distributed environments. In the previous section we presented an example of a web application, however such variations can happen in generally in any comprehensible application. An admired lead to this is to ensure scalability of the finishing system to be augmented to a size which was received to accommodate a compound stockpile.



**Figure 2.1 – Self-configuration control loop**

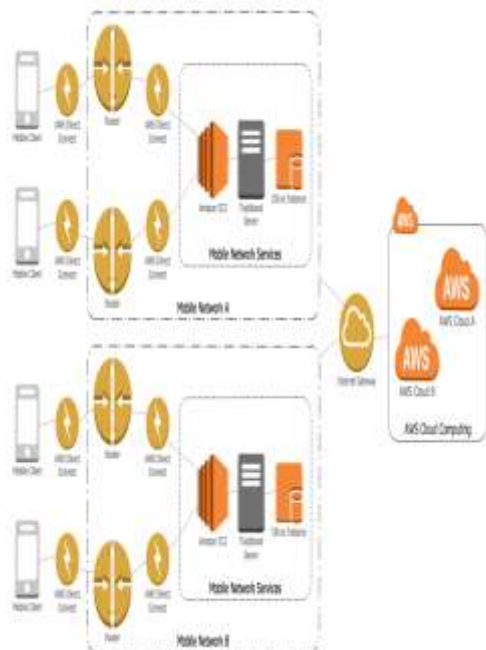
The disadvantage of such an entry is that the size of the system has to be estimated in the future. If the increase in the load does not agree to the estimation, the assigned ability is not used effectively. Another disadvantage is that scaling implies extending the resources, and it may be hard to scale the length after going on.

## II. SYSTEM ARCHITECTURE



**Fig 3.1: Architecture for elasticity in cloud computing**

The Amazon Web Services technology allows its subscribers to have full-fledged virtual computers. These computers are easy to use at any period to any of the Amazon users, who are showing off them, though they use the internet. The virtual PCs, which are delivered, have the permissible computer characteristics, such as a series of successful systems, networking and pre-loaded application software (for example, web servers, databases, etc.).



**Fig 3.2: Amazon Web Services Architecture**

In Amazon Web Services, each instance virtualizes its console, display mouse and keyboard. This console allows its users to use Amazon Web Services system using their browser. The users of AWS can log-in and use their virtual computers as their legitimate computers. The users can also pick to deploy their AWS systems in order to give much needed internet-based services as their customers lead.

This AWS technology is based upon a single virtual AWS computer, a real computer or even a few alternating real computers. Amazon keeps its right to re model, have enough maintenance and run the industry providing security to each of its subscribers system. The AWS services can be found in many global geographical regions and they add together as one of the services named Amazon Elastic Compute Cloud.

### III. ALGORITHM

#### RESOURCE AWARE ALGORITHM

The resource-aware formula methodology [5] has been developed for dominant the behavior of algorithms execution on devices with restricted resources, e.g. mobile devices or device nodes, and wherever the resources like memory, process cycles, communication information measure and battery life might degrade with times. The approach permits adapting however a formula handles its input, output and/or process functions supported out there resources. 3 key methods may be used:

- 1) Dominant the algorithm's Input graininess, e.g. by ever-changing the resolution or details of the input file structures;
- 2) Dominant the algorithm's process graininess, e.g. by performing arts less or a lot of computation and
- 3) Dominant the algorithm's Output graininess, e.g. by dominant the resolution or detail of the output knowledge structures.

The resource-aware algorithms methodology needs the implementation of a resource monitor and a choice mechanism (e.g. set of rules) for selecting between the 3 methods and permitting the formula to form trade-offs between

the standard of output result and also the out there resources. Though the approach builds implicitly on knowing however the standard of results varies with resources, it doesn't essentially need the definition of a particular quality operate.

#### IV. MODULE DIAGRAMS

##### ➤ 5.1 USE CASE DIAGRAM



**EC2 – Elastic Compute Cloud** blazeclean

EC2 is a Web Service that provides resizable Compute Capacity in the Cloud.

**PRICING**

Pay As You Go

- On-Demand
- Reserved Instance
- Spot Instances

**PRICING PARAMETERS**

- Region
- EC2 Instance type
  - vCPU
  - ECU
  - Memory (GB)
  - Instance Storage (GB)
  - Operating System

**FREE TIER**

- 750 hours Linux / UNIX, RHEL, SLES
- 750 hours Windows

**30 TYPES OF INSTANCES**

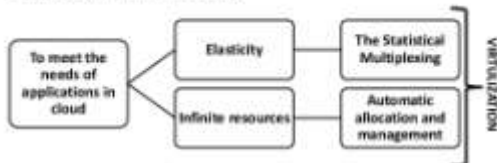
- Linux – Ubuntu, Red Hat Enterprise Linux, SUSE Linux
- Windows, Windows SQL Standard, Windows with SQL Web

EC2 Auto Scaling ELB IAM IAM Roles Direct Connect Amazon S3 AWS IoT AWS IoT Core AWS IoT Fleet Hub AWS IoT Jobs AWS IoT Rules Engine AWS IoT SiteWise AWS IoT Things Gateway AWS IoT Things Graph AWS IoT Things Studio AWS IoT TwinMaker AWS IoT XRay AWS IoT XRay Data Processor AWS IoT XRay Data Store AWS IoT XRay Data Visualizer AWS IoT XRay Data Warehouse AWS IoT XRay Data Visualizer AWS IoT XRay Data Warehouse AWS IoT XRay Data Visualizer

##### ➤ 5.2 CLASS DIAGRAM

#### Classes of Utility Computing

- Computation, Storage, Communication Model for an application is necessary.



- Utility Computing offerings will be distinguished based on
  1. the cloud system software's level of abstraction
  2. the level of management of resources

##### ➤ 5.3 ACTIVITY DIAGRAM



**Lightweight Elasticity**

Provisioning **on-demand** and not for peak  
**Optimize operating cost!**

Resources vs Time

**Traditional Infrastructures:** Shows a constant capacity line with a fluctuating demand line. The area between the demand line and the capacity line is shaded grey, representing unused resources.

**Deployment in the Cloud:** Shows a capacity line that dynamically adjusts to the demand line. The area between the demand line and the capacity line is shaded light blue, representing optimized resource usage.

Legend:  Unused resources

Source: [www.cs.cmu.edu/~jshreeraj/](http://www.cs.cmu.edu/~jshreeraj/) UCSB

##### ➤ 5.4 SEQUENCE DIAGRAM





```

22
23+ protected void doPost(HttpServletRequest request, HttpServletResponse
24
25     try {
26         Class.forName("com.mysql.jdbc.Driver");
27         Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/vivek");
28         PrintWriter pw=response.getWriter();
29         response.setContentType("text/html");
30         String uname=request.getParameter("uname");
31         String pass=request.getParameter("pass");
32         // pw.println("table created successfully");
33         String sql="insert into vivek values(?,?)";
34         PreparedStatement pst=con.prepareStatement(sql);
35         pst.setString(1, uname);
36         pst.setString(2, pass);
37         pst.executeUpdate();
38         pw.println("Record inserted successfully");
39
40
41     } catch (ClassNotFoundException | SQLException e) {
42         // TODO Auto-generated catch block
43         e.printStackTrace();
44     }
45
46
47
48

```

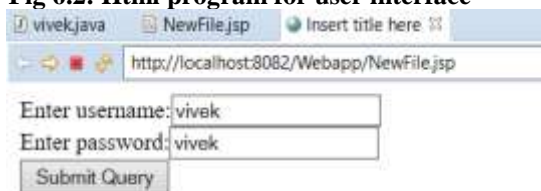
**Fig 6.1: A java program**

```

1 <!-- page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"%>
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <form action="vivek" method="post">
11 Enter username:<input type="text" name="uname"><br>
12 Enter password:<input type="text" name="pass"><br>
13 <input type="submit">
14 </form>
15
16 </body>
17 </html>

```

**Fig 6.2: Html program for user interface**



**Fig 6.3: User interface**

You will get a similar screen as above. Create a table in my sql command line client to save user input. Then type the details in html page. Your details will be saved in the table as shown below.

```

MySQL 5.7 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.21-log MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from vivek;
ERROR 1046 (30000): No database selected
mysql> use mysql;
Database changed
mysql> select * from vivek;
+-----+-----+
| uname | pass |
+-----+-----+
| vivek | vivek |
+-----+-----+
1 row in set (0.00 sec)

mysql>
    
```

**Fig 6.4: Saved user details in a table**

Output might not be same for everyone.

**Figures**

S.No	Name	Page No.
1.1	Comparison of classical scalability and elasticity approaches	02
2.1	Self-configuration control loop	03
3.1	Architecture for Elasticity in cloud Computing	04
3.2	Amazon Web Services Architecture	05
5.1	Use Case Diagram	06
5.2	Class Diagram	07
5.3	Activity Diagram	07
5.4	Sequence Diagram	07
5.5	Collaboration Diagram	07
5.6	Data Flow Diagram	08
6.1	A Java Program	09
6.2	HTML Program for User Interface	09
6.3	User Interface	10
6.4	Saved User Details in a table	10

**VI. CONCLUSION**

In this paper, we conduct a systematic literature review to recommend scalability, elasticity, and efficiency definitions and metrics for the cloud computing context. Based on our findings, we especially highlight commonly used concepts and propose novel definitions tied to cloud computing. Software architects can use our metric recommendations to analyse the quality of cloud applications and to derive new metrics based on the concepts we highlight. Cloud

providers and cloud consumers can use these concepts as common vocabulary and specify service level objectives based on our metric suggestions, without missing important concepts. For future work, we suggest to externally execute our SLR, thus, analyzing its reproducibility. Such re-executions can also vary parameter of our SLR to cope with some of our threats to validity. For example, different search databases, more synonyms, and a larger time frame can be considered. Such efforts can strengthen our results and/or add further concepts to be considered.

There are many more players in the upon-demand insist that many reports understand. These range from basic infrastructure offerings (IaaS), through platform preserve (PaaS) to full applications (SaaS). The long term cost of ownership may at first not seem to amass taking place, but endure into consideration new factors such as shortened risk and auxiliary value and for many organisations upon-demand facilities make a lot of wisdom.

Cloud computing is varying the exaggeration IT departments make a get your hands on of IT. Businesses have a range of paths to the cloud, including infrastructure, platforms and applications that are nearby from cloud providers as online facilities. Many people may be mortified by the range of offerings and the terminology used to describe them and will be unsure of the risk and foster.

## VII. REFERENCES

1. *Cloud Computing Principles and Paradigms*, John Wiley and Sons, 2011, [ISBN 978-0-470-88799-8](#)
2. Gambi, Alessio; Daniel Moldovan; Georgiana Copil; Hong-Linh Truong; Schahram Dustdar (2013). "[On estimating actuation delays in elastic computing systems](#)". *Proceedings of ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. doi:[10.1109/SEAMS.2013.6595490](#).
3. Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, Schahram Dustdar, "[SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications](#)", *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 14–16, 2013, Delft, the Netherlands
4. Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, Schahram Dustdar, "[Specifying, Monitoring, and Controlling Elasticity of Cloud Services](#)", *Proceedings of the 11th International Conference on Service Oriented Computing*. Berlin, Germany, 2–5 December 2013. doi=[10.1007/978-3-642-45005-1\\_31](#)
5. Herbst, Nikolas; Samuel Konev; Ralf Reussner (2013). "[Elasticity in Cloud Computing: What It Is, and What It Is Not](#)" (PDF). *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013)*, San Jose, CA, June 24–28.
6. Kranas, Pavlos (2012). "[ElaaS: An Innovative Elasticity as a Service Framework for Dynamic Management across the Cloud Stack Layers](#)". *Proceedings of Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS) 4–6 July 2012*. IEEE. doi:[10.1109/CISIS.2012.117](#).
7. Mao, Ming; M. Humphrey (2012). "[A Performance Study on the VM Startup Time in the Cloud](#)". *Proceedings of 2012 IEEE 5th International Conference on Cloud Computing (Cloud2012)*: 423. [ISBN 978-1-4673-2892-0](#). doi:[10.1109/CLOUD.2012.103](#).
8. Mencagli, Gabriele; Vanneschi, Marco (6 February 2014). "Towards a systematic approach to the dynamic adaptation of structured parallel computations using model predictive control". *Cluster Computing*. **17** (4): 1443–1463. doi:[10.1007/s10586-014-0346-3](#).
9. Moldovan, Daniel; Georgiana Copil; Hong-Linh Truong; Schahram Dustdar (2013). "[MELA: Monitoring and Analyzing Elasticity of Cloud Services](#)". *Proceedings of IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom 2013)*. doi:[10.1109/CloudCom.2013.18](#).
10. Nikolas Herbst, Rouen Krebs, Giorgio's Oikonomou, George Kousiouris, Athanasia Evangelinou, Alexandru Iosup, and Samuel Kounev. Ready for Rain? A View from SPEC Research on the Future of Cloud Metrics. Technical Report SPEC-RG-2016-01, SPEC Research Group - Cloud Working Group, Standard Performance Evaluation Corporation (SPEC), 2016. [\[1\]](#)
11. Perez; et al., *Responsive Elastic Computing*, [ISBN 978-1-60558-578-9](#)