International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0258
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1761 - 1765

Predictive Auto-Scaling Framework for Preventing Server Overload in High-Traffic Web Applications

Naveen Kumar.M[1], Prajwal Nayaka N. S[2], Prajwal Gajanan Acharya[3], Sudeep S[4], Sushma S[5], Dr Kumudavalli M.V [6]

[1,2,3,4,5]PG Student, Department of MCA, [Dayanand Sagar College of Arts, Science and Commerce], Karnataka, India.

[6]Professor, Department of MCA, [Dayanand Sagar College of Arts, Science and Commerce], Karnataka, India.

EmailID: *naveenkumarm1667@gmail.com[1] ,sushmasomanna2003@gmail.com[2], sudeecta.sudeep69@gmail.com[3], prajwalnayaka686262@gmail.com[4], prajwalacharya.810@gmail.com[5]*

Abstract

Many web systems, such as university portals, banking platforms, and ticket booking websites, sometimes show a server busy error or experience server crashes when many users try to access the system at the same time. During peak hours server is not able to handle all requests properly. Because of that users get slow response and sometimes service interruptions also happens. So users face problems and trust on system also getting reduced.In this work a predictive auto scaling framework is designed to reduce server busy issues and make website more stable when many users are using at same time. Idea is simple users should not face server problems even if traffic suddenly increases. Here a cloud based approach is used. Number of users and incoming requests are monitored continuously. When traffic increase is predicted extra virtual servers are prepared before and workload is distributed across different machines so load can be handled better. As a result system performance is improved and chances of server crash are reduced. Overall heavy traffic can be handled in better way and users will get better service.

Keywords: *Auto-scaling; Cloud Computing; Load Balancing; Server Overload; Traffic Prediction*

1. Introduction

In recent years, web based systems like university portals, banking apps and e-commerce sites have grown a lot in user traffic. During peak times like result announcements or flash sales, sudden increase in requests are not handled well and server busy errors or crashes happens. Because of this, user experience gets affected and trust on the system also reduces.Traditional methods like upgrading hardware or increasing server capacity manually are used to solve this problem. But these methods are costly and not very efficient, and also they cant handle sudden changes in traffic properly. With cloud computing, scalable infrastructure is available where resources can be adjusted based on demand. In this research, a predictive auto scaling framework is designed by using artificial intelligence, cloud computing and load balancing. The system reliability is improved, response time is reduced and server overload during high traffic conditions are tried to be prevented.

1.1.Problem Statement

In this study, the main issue being focused is that existing systems are not able to manage sudden rise in user traffic properly. [1-2] Because of this, many problems are getting created such as:

- Server capacity is limited and gets overloaded easily
- Response from the system becomes slow or delayed
- Resources are not used in an efficient way

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0258
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1761 - 1765

- Real time scaling is not available in most cases

1.2.Objectives

This research mainly aims to achieve the following points:

- To build a cloud based system which can handle more users smoothly
- To estimate future traffic by using AI based methods
- To distribute the load in a better and balanced way
- To minimize system downtime and avoid failures as much as possible

2. Method

In this section, a method is described for handling high traffic in web applications by using a cloud based predictive scaling system. The approach mainly combines three parts such as AI based traffic estimation, cloud resource handling and load sharing between servers. At the beginning, the application is deployed on a cloud environment where many virtual servers are available. Between the user and servers, a load balancer is placed so that incoming requests are shared across servers. By doing this, one single server is not getting too much load during peak time.[3-4]
Further, an AI module is included to study past traffic data like user behaviour, peak hours and request count. From this data, future traffic is estimated and system gets ready before the actual load increases. Because of this, sudden overload problem can be avoided to some extent. Cloud scaling mechanism is also used where number of servers are adjusted automatically. When more traffic is expected or detected, extra servers are added. In the same way, when traffic becomes less, unused servers are removed so that cost and resources are not wasted.
Also, a monitoring system is running continuously to check important parameters like CPU usage, memory usage, response delay and number of users. If these values goes beyond a limit, scaling action is triggered automatically. This helps in keeping system performance stable most of the time.

Table 1 Experimental input parameters for Auto scaling Framework

Parameter	Range / Value	Explanation
CPU Usage (%)	60 – 80	At this level, system may start scaling
Memory Usage (%)	65 – 85	Shows how much memory is being used currently
Active Users	1,000 – 50,000	Total users using system at the same time
Response Time (ms)	100 – 2000	Time taken by system to give response
Scaling Type	Add / Remove	Servers are increased or decreased based on need
Prediction Interval	5 – 10 minutes	Time gap used for checking and predicting traffic
Load Balancing Method	Round Robin	Requests are shared one by one to servers

2.1.Tables

Tables are used to present the system details in a simple and organized manner. Each table is numbered and provided with a short explanation. Table 1 includes the main parameters that help in deciding when scaling is needed and how the system performs under different traffic conditions.[5]

2.2.Figures

Figures are used to explain the working of the system in visual form. Each figure is numbered properly In this design, users are sending requests to the system, and those requests are first handled by a load balancer. Then requests are divided among multiple servers. These servers process the data and connect with database when needed. At the same time, AI module is checking previous traffic and predicting future load. Based on this, cloud system increases or decreases number of servers. Monitoring part is

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0258
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1761 - 1765

always checking system condition and takes action when required. This full setup helps in reducing server busy problem and improves overall performance.

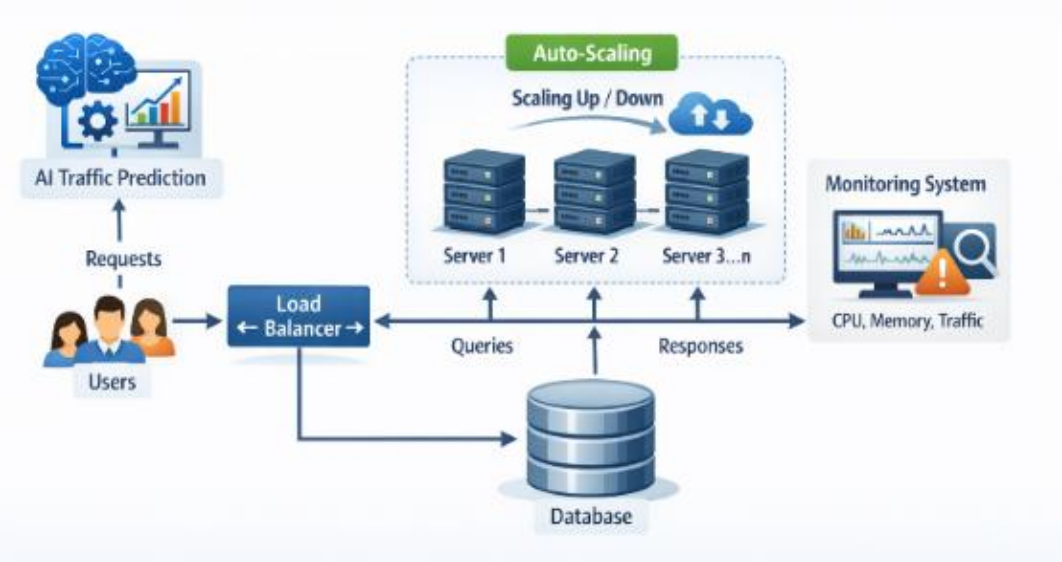

Figure 1 Proposed Cloud-Based Auto-Scaling Architecture Flow

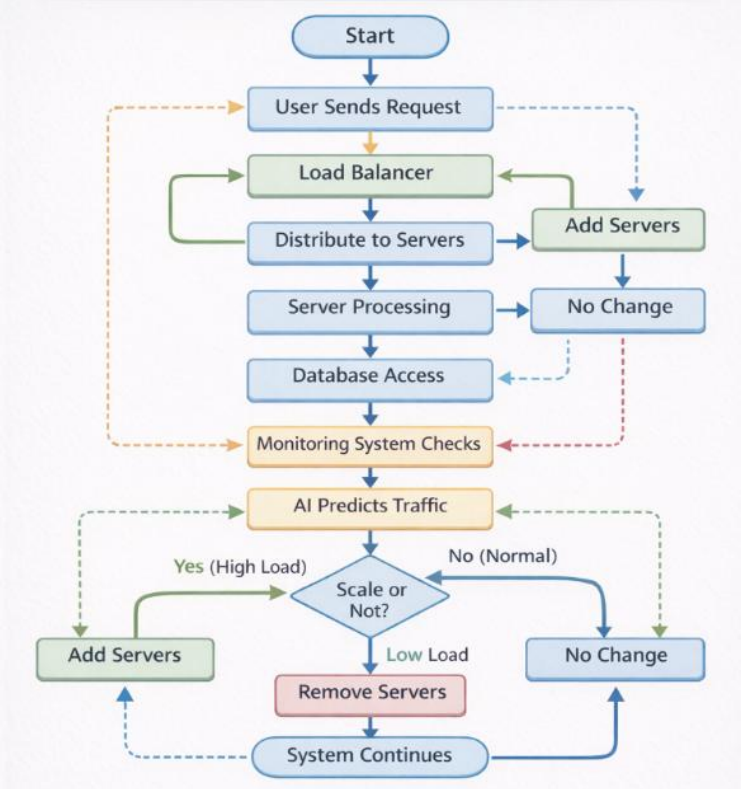

Figure 2. Workflow of Proposed Auto – Scaling System

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0258
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1761 - 1765

The workflow of this system is basically how the whole process happens step by step when users use the application. First, users send their requests to the system. These requests don't go directly to one server, instead they go to a load balancer. The load balancer will split the requests and send them to different servers, so that no single server gets too much load. After that, each server will handle the request and take needed data from the database. Once the work is done, the response is sent back to the user. At the same time, the system is also watching what is happening inside, like how much CPU is used, memory usage, response time and how many users are active. [5] There is also one prediction part which checks old traffic data and tries to guess future traffic. It is not always perfect, but it gives some idea. Based on this and current condition, system decides whether to increase servers or keep it same. If suddenly more users come, then new servers are added so system can handle it. And when users reduce, extra servers are removed so resources are not wasted. This cycle keeps running again and again. Because of this, system works more smoothly and chances of server busy problem becomes less.

3. Results and Discussion

3.1.Results

The cloud based predictive auto scaling system was tested in different traffic conditions to check how it manage server busy problems. For testing, users were increased from nearly 1,000 to 50,000 at same time.
From what we observed, system was working better than normal single server or basic scaling type systems. When traffic goes high, extra servers are getting added automatically, so requests are handled more smoothly. Response time was almost same even when load becomes high, mainly because load balancing and prediction both are working together.
Some things we noticed:

- Response time was going up slowly, not suddenly during peak
- System handled more number of users without much problem
- Load was divided properly between servers
- No major crash or server busy error happened
- But in normal systems without auto scaling, response time becomes very high and sometimes system stops working when users cross the limit.

3.2.Discussion

- From these results, we can understand that using AI prediction with cloud scaling and load balancing is more better way to handle high traffic. Instead of waiting for problem, system is trying to prepare before by predicting traffic. [6]
- Load balancing helps in sharing requests between servers, so one server will not get overload. Cloud scaling increases or decreases servers based on need, so cost also can be reduced when traffic is less.
- There are some drawbacks also. Prediction depends on previous data, so if traffic suddenly changes, it may not give correct result and small delay can happen. Also system setup is not very easy, it needs proper configuration and monitoring.
- Still, this system can be useful in real cases like university portals, banking systems and e-commerce websites. It helps in reducing downtime, improving response and overall it gives better experience to users.[6] .

Conclusion

This research mainly focused on the problem of server busy issues in high traffic web applications. These problems usually causes slow response, system crash and bad user experience. From the results and discussion, it is clear that traditional systems are not able to handle sudden increase in users properly. The proposed cloud based predictive auto scaling framework gives a better solution by using AI, cloud computing and load balancing together. The system can predict traffic before it happens, allocate resources when needed and also distribute requests between multiple servers. Because of this, server overload is reduced and system works more stable even during peak time. From the testing, it was seen that the system gives lower response time, better load sharing and very less failure when compared to normal methods. It also uses resources in better way by adding or removing servers based on demand. Finally, this system can be used in real applications like university portals, banking systems and e-

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0258
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1761 - 1765

commerce websites. It is scalable, reliable and also cost effective. In future, prediction part can be improved more by using better machine learning models.

Acknowledgements

We are thankful to the Department of MCA, Dayananda Sagar College of Arts, Science and Commerce, Bengaluru for giving us the support to do this project. Without their help, it would have been difficult to complete the work. We also want to thank our teachers and mentors for helping us whenever we had doubts and for guiding us time to time. Their support really helped us to finish this project successfully.

References

[1]. Armbrust M., Fox A., Griffith R., Joseph A.D., Katz R., Konwinski A., Zaharia M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58

[2]. Lorido-Botran T., Miguel-Alonso J., Lozano J.A. (2014). Auto scaling techniques for elastic applications in cloud environments. University of Basque Country

[3]. Mao M., Li J., Humphrey M. (2016). Cloud auto scaling with deadline and budget constraints. IEEE International Conference on Cloud Computing, 41-48

[4]. Calheiros R.N., Ranjan R., Beloglazov A., De Rose C.A., Buyya R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments. Software Practice and Experience, 41(1), 23-50

[5]. Kratzke N., Quint P. (2017). Understanding cloud native applications after 10 years of cloud computing. Journal of Systems and Software, 126, 1-16

[6]. Xu J., Fortes J.A. (2010). Multi objective virtual machine placement in virtualized data center environments. IEEE International Conference on Green Computing, 179-188