

Self Chord-Achieving Load Balancing In Peer To Peer Network

M.Divya¹ B.Saranya²

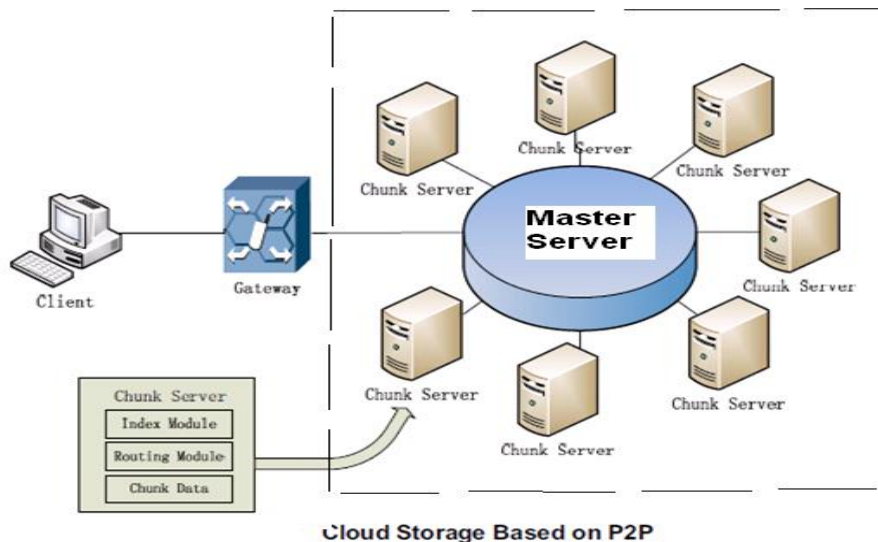
Lecturer, Department Of Cse, As-Salam College Of Engg And Tech

Abstract:- The Cloud computing technology has been widely applied in e-business, e-education. Cloud computing platform is a set of Scalable large-scale data server clusters, it provides computing and storage services to customers. The cloud storage is a relatively basic and widely applied service which can provide users with stable, massive data storage space. Our research shows that the architecture of current Cloud Computing System is central structured one; all the data nodes must be indexed by a master server which may become bottle neck of the system. In this project we use cloud storage architecture based on P2P, and balance CPU load on server.

A Distributed Hash Table method for storing hash tables in geographically distributed locations in order to provide a failsafe lookup mechanism for distributed computing. Various algorithms have been explored that provide the right balance and speed for storing parts of the tables in different locations. Having been used in the early and mid-1990s for local area network storage, after the turn of the century, DHTs were brought into focus for peer-to-peer computing over the Internet. A DHT provides a fault tolerant storage interface on top of which is layered an application such as music sharing, file sharing or distributed backup.

I. INTRODUCTION

Computing systems, such as computational Grids and Clouds are an information service that provides information about the resources that can be used to build and run complex applications and enables their discovery. Grids use the resources of many networked computers to solve large-scale computation problems in multiple and heterogeneous domains. The large-scale and dynamic nature of Grids make human administrative intervention difficult or even unfeasible, and centralized information services are proving unsuitable to scale to hundreds or thousands of nodes. To tackle these issues, the scientific community has proposed to design information services according to the peer-to-peer (P2P) paradigm, which offers better scalability and adaptivity features. A similar trend can be envisioned for the recently emerged Cloud paradigm, which is switching computation and storage responsibilities from the client side to the “clouds,” i.e., to unseen computers on the server side, possibly scattered across continents. Grid and Cloud issues are similar in many aspects, especially in the need to assure scalability in a dynamic environment. Therefore, P2P techniques are very likely to be adopted in Clouds as they are today in Grids. Self-Chord, a P2P system that inherits from Chord the ability to construct and maintain a structured ring of peers, but features enhanced functionalities achieved through the activity of ant-inspired mobile agents. In particular, Self-Chord features the following benefits with respect to Chord.



- 1) In Self-Chord, there is no obligation to assign a key to a well-specified peer. A user can issue “class” queries, i.e., explore the network to find resources belonging to a specified class and then select the most appropriate for his/her purpose.
- 2) Structured systems like Chord can produce imbalance problems depending on the location of peers and the statistical distribution of the values of resource keys. In Self-Chord, the keys are fairly distributed over the peers, irrespective of the location of peers and the distribution of key values.
- 3) In Chord, appropriate operations are necessary when a peer joins the ring or when new resources are published. These resources must be immediately assigned to the peers whose indexes match the resource keys. These operations are not necessary in Self-Chord. This assures scalability and robustness with respect to environmental changes.

II. IMPLEMENTATION

PROJECT OBJECTIVE

We propose a cloud computing architecture based on P2P which provide a pure distributed data storage environment without any central entity. The cloud based on the proposed architecture is self-organized and self-managed and has better scalability and fault tolerance. Here we develop architecture with cloud storage mechanism based on P2P network. Hence the load of CPU server is balanced, with better scalability and fault tolerance.

III. ALGORITHMS AND TECHNIQUES USED

DHT IMPLEMENTATIONS

The DHT chord algorithm is used in the searching process for our project for its efficiency.

- Let the current node k, be first node in question.
- Let successor be the next node from the node in question on the identifier ring.
- Let predecessor be the previous node from the node in question on the identifier ring.
- K finds predecessor and sends data.
- Else it sends data to its successor.
- Follow last two steps till data reaches the destination. Stop the process.
- The Distributed Hash table will consists of the following:

hostname	predecessor	Successor	cpuval
192.168.1.6	192.168.1.5	192.168.1.4	5
192.168.1.4	192.168.1.6	192.168.1.5	6
192.168.1.5	192.168.1.4	192.168.1.6	2

Distributed Hash Table

IV. MODULE DESCRIPTION

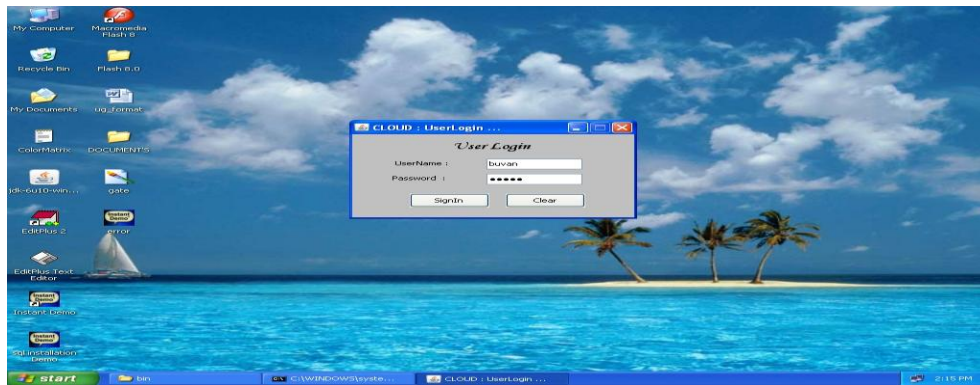
- Client
- Gateway
- Chunk server

1. Client

The client application which wants to get the data from the platform.

Uname	pwd
Archana	confidence
Buvani	winner
Dhivyaa	success

Client

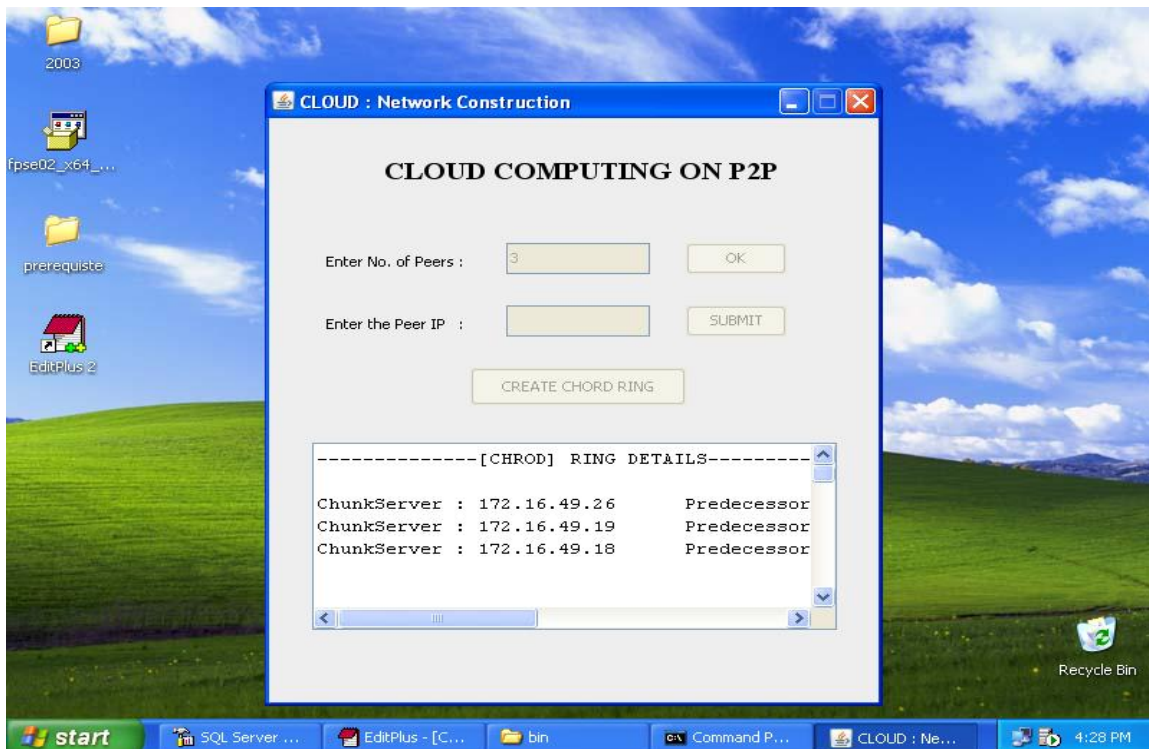


2.Gateway

The entity which can transfer the request or response between the Client App with the network and can lead the request to the nearest node in the network. The logicid table consists of these items:

Fname	logicid
1.txt	dd7ec931179c4dcb6a8ffb8b8786d20b
2.txt	c3d57eb88086a04b1e04d06a9b6188e5
Blank.txt	6ba0adabe46128552ccd513f2d40f250
help.txt	6477fc60281b47a0f905549f657bd98c
README.txt	26fd799ea07494916e9da9b91b2aac64
RUNNING.txt	e3888c78ec581348ba38ea09f87d8644

Table.5.3.2 Gateway



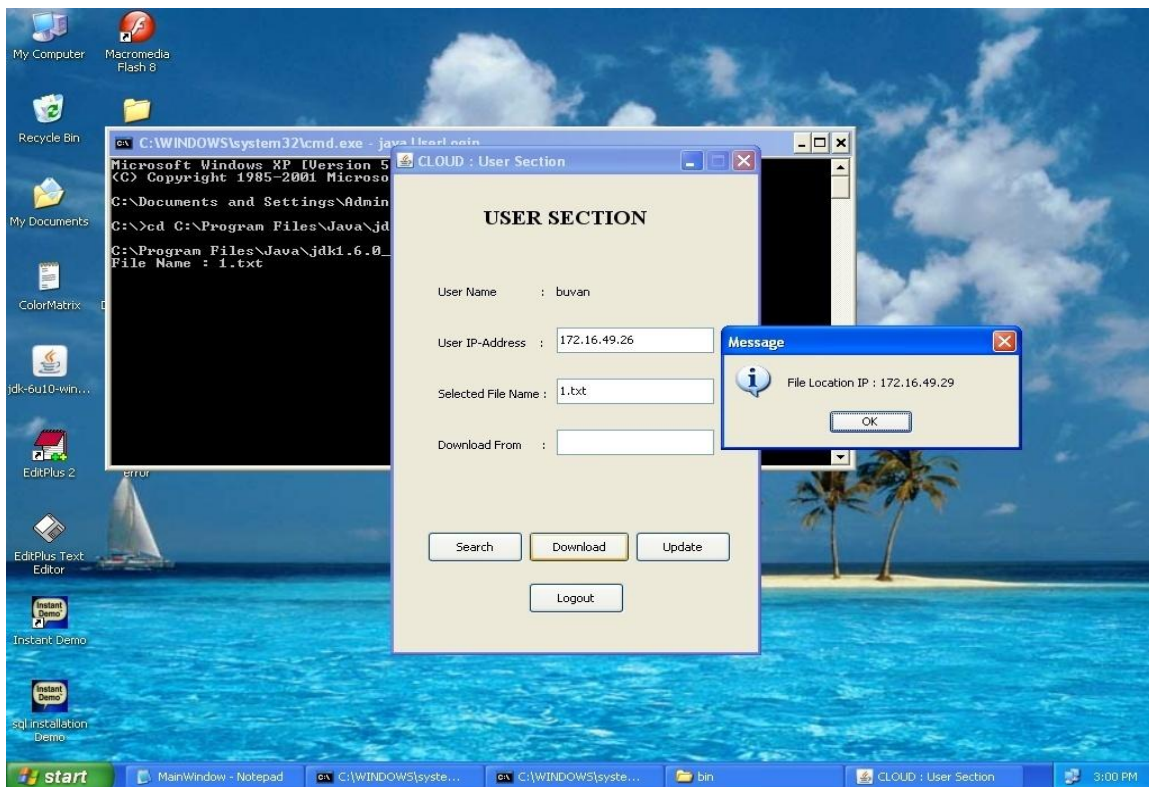
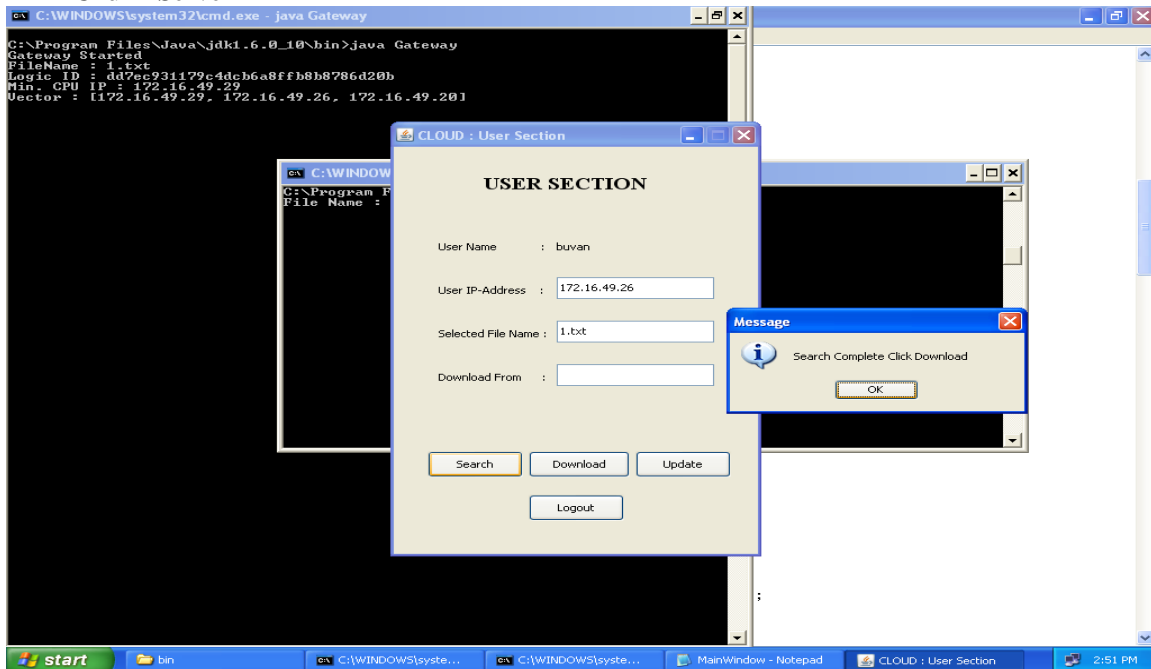
3.Chunk Server

The entity which is served as the data resource node and P2P node. Different with the function of pure data storage in GFS, the chunk server here has three function modules with separated interfaces.

Index Module, take charge of part of the global resource index which is assigned by DHT arithmetic such as Chord, Pastry and so on. Route Module, pass a lookup request by a next hop routing table which is also assigned by DHT. Data Module, provide the data resource stored in the local machine. The Chunk server will consists of the following information:

url	logicid
192.168.1.4	e3888c78ec581348ba38ea09f87d8644
192.168.1.6	6ba0adabe46128552ccd513f2d40f250

Chunk Server



V. RESULTS AND DISCUSSION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. Cloud Services has better scalability and dynamic behavior, using these services we can achieve load balancing in Peer to Peer network.

Self chord offers functionalities such as: Better support of complex discovery requests: In Self-Chord, the definition of resource keys is flexible and uncorrelated with peer indexes, and it is also possible to give a semantic meaning to key values. This enables the system to serve “class” queries, issued to search for resources having common characteristics.

Better balance of storage load: Self-Chord improves the balance of storage load among peers improved dynamic behavior: Self-Chord decouples the naming of resources and peers, resulting in two sets of keys/indices that can have different cardinalities. Self-Chord focuses on the real objective, which is the reordering of keys over the ring and their fair distribution to the peers. Self-Chord does not need any superstructure to achieve a fair load balance.

VI. FUTURE ENHANCEMENTS

Using the load balancer cloudblock also makes it simple to expand or smoothly transition your physical servers to the cloud service. To use the cloud for expansion, take a load balancer cloudblock and use it to route the traffic to one or more physical servers. When you need more capacity, turn up a virtual web server cloudblock, upload your web site to it, and add it to the load balancer.

REFERENCES

JOURNALS

- [1] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” *ACM Comput. Surveys*, vol. 36, no. 4, pp. 335–371, 2004.
- [2] A. Forestiero and C. Mastroianni, “A swarm algorithm for a self-structured P2P information system,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 681–694, Aug. 2009. the Italian National Research Council, ICAR-CNR, Cosenza, Italy, since 2003.
- [3] B. Hayes, “Cloud computing,” *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008P

CONFERENCE PROCEEDINGS AND CONFERENCE PAPERS

- [4] O. Babaoglu, H. Meling, and A. Montresor, “Anthill: A framework for the development of agent-based peer-to-peer systems,” in *Proceedings of 22nd ICDCS*, Washington, DC, 2002, pp. 15–22.
- [5] H. Balakrishnan, I. Stoica, R. Morris, D. Karger, and M. F. Kaashoek, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *Proceedings of ACM SIGCOMM*, San Diego, CA, 2001, pp. 149–160.
- [6] Francis, M. Handley, R. Karp, S. Ratnasamy, and S. Schenker, “A scalable content-addressable network,” in *Proceedings of ACM SIGCOMM*, San Diego, CA, 2001, pp. 161–172.
- [7] A. Forestiero, C. Mastroianni, and M. Meo, “Self-chord: A bio-inspired algorithm for structured P2P systems,” in *Proceedings of 9th IEEE CCGrid*, May 2009, pp. 44–51.
- [8] I. Foster, S. Lu, I. Raicu, and Y. Zhao, “Cloud computing and grid computing 360-degree compared,” in *Proceedings of Austin, TX*, Nov. 2008, pp. 1–10.
- [9] K. Sycara, “Multiagent systems,” *Artif. Intell. Mag.*, vol. 10, no. 2, pp. 79–93, 1998. proceedings. He edited special issues for the journals *Future Generation*.

WEB REFERENCE

<http://www.dmst.aueb.gr/dds/pubs/jrnl/2004-ACMCS-p2p/html/AS04.pdf>

<http://www.cs.unibo.it/projects/anthill/papers/2001-09.pdf>

http://pdos.csail.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf

<http://dns2.icar.cnr.it/forestiero/pub/TEVC-Forestiero-Mastroianni-2009.pdf>