Reducing upload and Download Time on Cloud using Content Distribution Algorithm

S.Priyanka[1], R.Kalpana [2], M.Hemalatha[3]
[1]*First Year ME (CSE), Kathir college of Engineering, Coimbatore, Tamilnadu,India.*
[2]*First Year ME (CSE), Adhithya Institute of Technology,Coimbatore,Tamilnadu,India.*
[3]*First Year ME (VLSI Design), Regional Center of Anna University,Coimbatore,Tamilnadu,India.*
[1]*priyankazappy@gmail.com*, [2]*kalpu_vijayakumar13@yahoo.co.in*, [3]*hemalathamuthu8@gmail.com*

***Abstract*- Cloud computing is a term, which involves virtualization, distributed computing, networking, software and web services. Cloud services provide resources efficiently based on demand. Central of these lies in the establishment of an effective algorithm to Achieve minimum distribution time (MDT). Achieving MDT is crucial for bulk-synchronous applications, when every client in the set to finish their download before being able to make use of the downloaded content. In this paper, we propose the use of dedicated Servers to accelerate peer-assisted content distribution using content distribution algorithm. Downloading time is reduced using BitTorrent application and Steiner tree algorithm. BitTorrent, a popular Peer-to-Peer file sharing protocol for mass distributions. Steiner tree algorithm, a star based protocol to effectively reduce the distribution time.**

Keywords- BitTorrent; Steiner Tree Algorithm; Minimum Distribution Time; Content Distribution.

*****

I. INTRODUCTION

1.1 Overview:

Cloud computing is an emerging topic in the field of parallel and distributed computing. It is an internet based computing, whereby shared resources, software's, and information are provided to computers and other devices on demand like electricity grid. Central of these lies the establishment of an effective algorithm to achieve minimum distribution time (MDT).An MDT objective implies that the overarching common goal of the provider and clients is to minimize the time by which all clients finish their download. BitTorrent is a popular file sharing protocol that is in wide use today. It has been successful as a method of mass content distribution and makes up a large fraction of P2P traffic. The basic idea in BitTorrent is to divide up a large file into equal-sized pieces and peers (nodes interested in downloading the file) downloaded and upload these pieces to other simultaneously. It is a P2P application that capitalizes on the bandwidth of peers to effectively replicate contents on a large set of peers. A specialty of Bit Torrent is the notion of torrent, which defines a session of transfer of a single content to a set of peers. Torrents are independent. A torrent is alive as long as there is at least one copy of each piece in the torrent.

1.2 Problem Statement:

Content distribution has emerged as a critical application as demand for high fidelity multimedia content has soared. Large multimedia files require effective content distribution services. Past solutions to the content distribution problem can be categorized into approaches, namely client-server systems and peer-to-peer systems whose fundamental limitations render them inadequate for many development environments. In the client-server approach, the content Owner operates a set of servers that provide the content to every client without tapping into any client-side resources. The drawback is cost and feasibility. Peer-to-Peer systems offer an emerging alternative, where clients interested in downloading a file provide content to other clients interested in the same file.

1.3 Objectives:

The main goal is to distribute data and reduce the distribution time using content distribution algorithms.
1. Understanding Bit Torrent and Steiner Tree algorithm.
2. Write a simple Web based application and develop a system to achieve fast and minimum distribution time.
3. Host the Web application on the cloud.

II. RELATED WORK

A. BitTorrent Application:

BitTorrent, an efficient file sharing system, in this application the system throughput is capped by the minimum of the total system upload bandwidth and the total system download bandwidth. A large population of home internet users today have asymmetric internet service, such as ADSL
And cable, which makes the total available upload capacity of all downloading peers the bottleneck of a Bit Torrent like file sharing systems.
However, there are other users with spare upload capacity which are not interested in any particular file. We will term such nodes helpers. Helpers represent a rich resource of untapped upload bandwidth which can be exploited for increasing the total system upload bandwidth and hence easing the bottleneck. The idea of utilizing networks free upload capacity was first introduced by Wong. Tribler is a social based peer-to-peer system which provides a mechanism for providing incentive for these helpers to help other users.

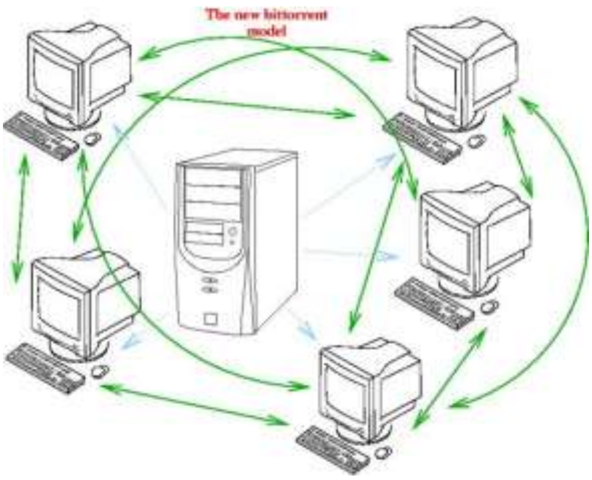
Our focus is on developing a systematic approach to optimally utilize such helpers.
We develop an efficient mechanism for utilizing the spare upload capacity of helpers while imposing very little download or storage burden on them. In our system, each helper only downloads a small number of random pieces of the file and actively searches for peers who do not possess these pieces to upload. This way they attempt to fully utilize their upload capacity. Our proposed design is a Bit Torrent- like system that is fully backward compatible with the original Bit Torrent protocol; existing Bit Torrent users can fully realize the gains from these helpers without any modification on their part. We provide an analysis of steady-state performance using a comparative study of both content distribution algorithms.

BITTORRENT ARCHITECTURE*:*

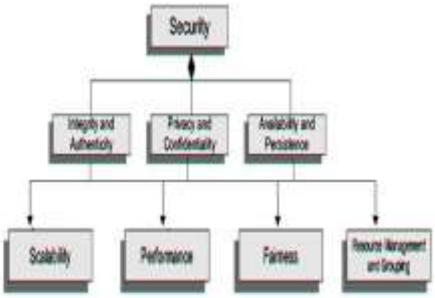
A BitTorrent client is a computer program that manages downloads and uploads using the Bit Torrent protocol. Bit Torrent is for distributing large amounts of data over the Internet. Bit Torrent is one of the most common protocols for transferring large files and it has been estimated that networks collectively have accounted for roughly 43% to 70% of all Internet Traffic. The Bit Torrent protocol can be used to reduce the server and network impact of distributing large files. Rather than downloading a file from a single source server, the Bit Torrent protocol allows users to join a file of hosts to download and upload from each other simultaneously. The protocol is an alternative to the older single source, multiple mirror sources technique for distributing data, and can work over networks with lower bandwidth so many small computers, like mobile phones, are able to efficiently distribute files to many files.

Pieces of data are typically downloaded non-sequentially and are rearranged into the correct order the BitTorrent Client, which monitors which pieces it has, can upload to other clients and which it needs. Pieces are of the same size throughout a single download (for example a 10MB file may be transmitted as ten 1MB Pieces or as forty 256kB Pieces). Due to the nature of this approach, the download of any file can be halted at any time and be resumed at a later date, without the loss of previously downloaded information, which in turn makes BitTorrent particularly useful in the transfer of larger files. This also enables the client to seek out readily available pieces and download them immediately, rather than halting the download and waiting for the next file which reduces the overall download of the file.

- A BitTorrent client is any program that implements the Bit Torrent protocol. Each client is capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A file is any computer running an instance of a client.
- To share a file or group of files, a client first creates a small file called Torrent. This file contains Meta data about the files to be shared and about the computer that coordinates the file distribution. Clients that want to download the file must first obtain a torrent file for it and connect to the clients to download the pieces of the file. Bit Torrent makes many small data requests over different cloud connections to different machines, while classic downloading is typically made via a single LAN connection to a single machine.
- BitTorrent downloads in a random or in a Split File approach that ensures high availability, while classic downloads are sequential.

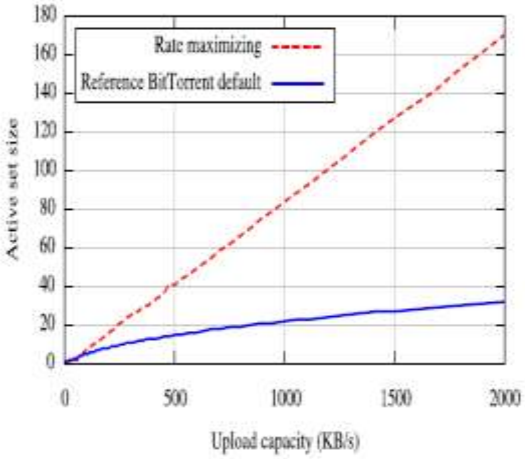


In bittorrent data is obtained from the other clients downloading at that time. Obviously the clients also are downloading at that time and they don't have all the bits but they share whatever they have. The key is that each client downloads a different piece of data and thus any new client can download different data from different clients. The most interesting thing here being that as more and more clients connect for downloading at the same time and hence the load gets distributed amongst themselves. Hence instead of loading the server, the load is distributed.


Security
Integrity and Authenticity
Privacy and Confidentiality
Availability and Persistence
Scalability
Performance
Fairness
Resource Management and Grouping

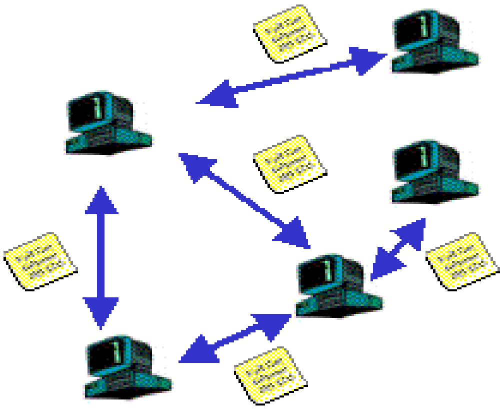
BitTorrent Upload related File:

- BitTorrent makes many small data requests over different TCP connections to different machines, while classic downloading is typically made via a single TCP connection to a single machine.
- BitTorrent downloads in a random or in a "rarest- first" approach that ensures high availability, while classic downloads are sequential.



A. CONTENT DISTRIBUTION

Content distribution on the Internet uses many different service architectures, ranging from centralized client-server to fully distribute. The recent wide-spread use of peer-to-peer applications such as SETI, Napster, and Gnutella indicate that there are many potential benefits to fully distributed peer-to- peer systems. Peer-to-peer content distribution provides more resilience and higher availability through wide-scale replication of content at large numbers of peers.



The boxes in the periphery show the most important attributes of content distribution systems, namely:

Integrity and Authenticity: Safeguarding the accuracy and completeness of data and processing methods. Unauthorized entities cannot change data; adversaries cannot substitute a forged document for a requested one.

Privacy and Confidentiality: Ensuring that data is accessible only to those authorized to have access and that there is control over what data is collected, how it is used and how it is maintained.

Availability and Persistence: Ensuring that authorized users

Have access to data and associated assets when required. For a content distribution system this often means always. This property entails stability in the presence of failure or changing node populations.

Performance: The time required for performing the operations allowed by the system, typically publication, searching and retrieval of documents.

Fairness: Ensuring that users offer and consume resources in a fair and balanced manner. May rely on accountability, reputation and resource trading mechanisms.

Resource Management Capabilities: In their most basic

Form peer-to-peer content distribution systems allow the publishing, searching and retrieval of documents. More sophisticated systems may afford more advanced resource management capabilities such as editing or removal of documents, management of storage space and operations on metadata.

Semantic Grouping of Information: An area of research that has attracted considerable attention recently is the semantic grouping and organization of content and information.

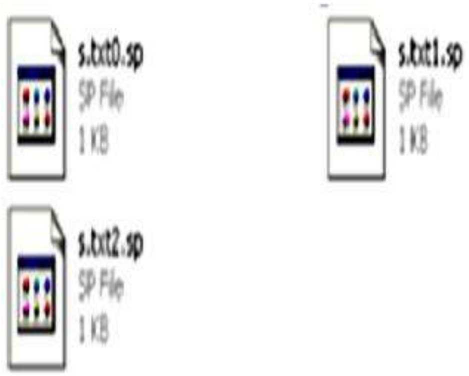
The problem identifies in the systems with a *Minimum Distribution Time* (MDT) objective. An MDT objective implies that the goal of the provider *and* clients is to minimize the time by which *all* clients finish their download. To achieve An MDT objective for a group of client requires judicious use of client uplink and downlink capacities, which are typically highly asymmetric as offer their clients, download speeds that are significantly larger than upload speeds. In support of the MDT objective for bulk-synchronous content distribution, in this paper we investigate the potential benefit from the on- demand deployment of cloud resources to upload capacity, we propose the use of Bit Torrent in receiving the entire content.

FEATURE:

- The Amazon S3 "Simple Storage Service" is a scalable Internet-based storage service with a simple web service interface, equipped with built-in BitTorrent support.
- The UK government used BitTorrent to distribute details about how the tax money of UK citizens was spent.

- BitTorrent has a widespread use in different fields like Films, Videos and Music.

Distributed File:



s.txt0.sp
SP File
1 KB

s.txt1.sp
SP File
1 KB

s.txt2.sp
SP File
1 KB

The uploaded file is split ted and stored in different nodes .

A. STEINER TREE CONTENT DISTRIBUTION

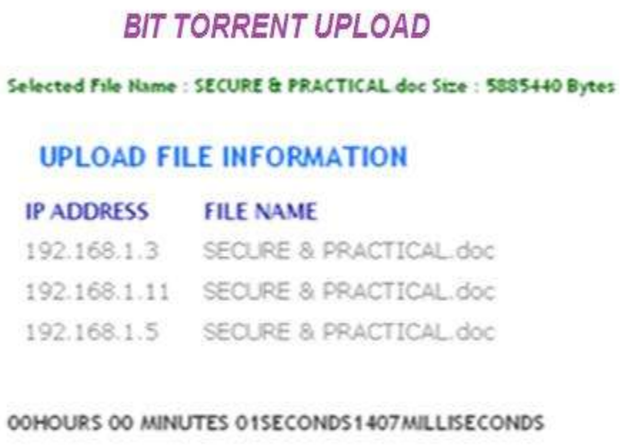
Steiner Tree content distribution algorithm is to

reduce
The upload and download time using a comparative work with Bit Torrent Application. The problem is to find a Steiner tree connecting the source node and destination nodes that minimizes data distribution time. To minimizing upload and download time the Algorithm implemented is BitTorrent and Steiner tree algorithm.

The uploaded file is splitted and stored in different nodes. The IP address and the splitted file name of the corresponding node will be displayed.

The uploaded file is split ted and stored in different nodes using Bit Torrent application shown in the figure.

Uploaded File Information:



BIT TORRENT UPLOAD

Selected File Name : SECURE & PRACTICAL.doc Size : 5885440 Bytes

UPLOAD FILE INFORMATION

IP ADDRESS	FILE NAME
192.168.1.3	SECURE & PRACTICAL.doc
192.168.1.11	SECURE & PRACTICAL.doc
192.168.1.5	SECURE & PRACTICAL.doc

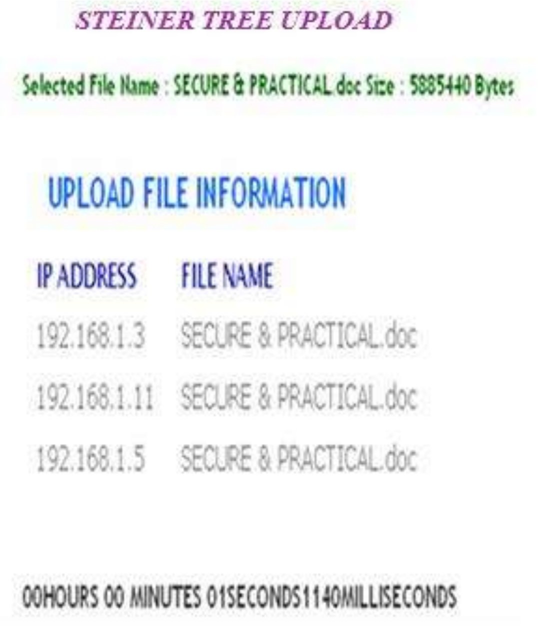
00HOURS 00 MINUTES 01SECONDS1407MILLISECONDS

The uploaded files are in three different peer nodes and in same size using Steiner Tree concept.

The uploaded file information will be stored in this folder. The folder name is same as the file name which will be created automatically during the uploading process.

The time is reduced here and that will be shown in the comparative work.

Downloaded File Information:



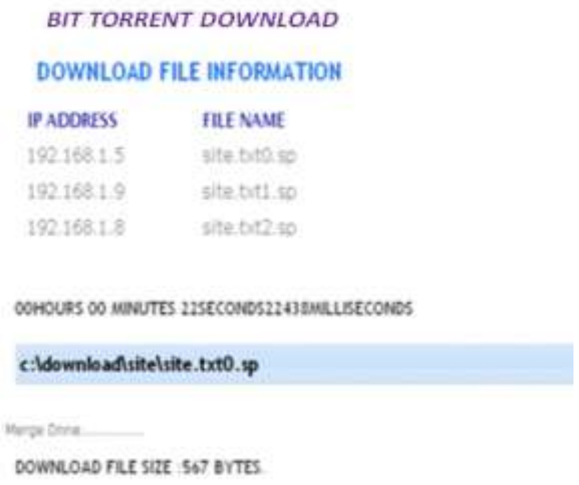
STEINER TREE UPLOAD

Selected File Name : SECURE & PRACTICAL.doc Size : 5885440 Bytes

UPLOAD FILE INFORMATION

IP ADDRESS	FILE NAME
192.168.1.3	SECURE & PRACTICAL.doc
192.168.1.11	SECURE & PRACTICAL.doc
192.168.1.5	SECURE & PRACTICAL.doc

00HOURS 00 MINUTES 01SECONDS1140MILLISECONDS



BIT TORRENT DOWNLOAD

DOWNLOAD FILE INFORMATION

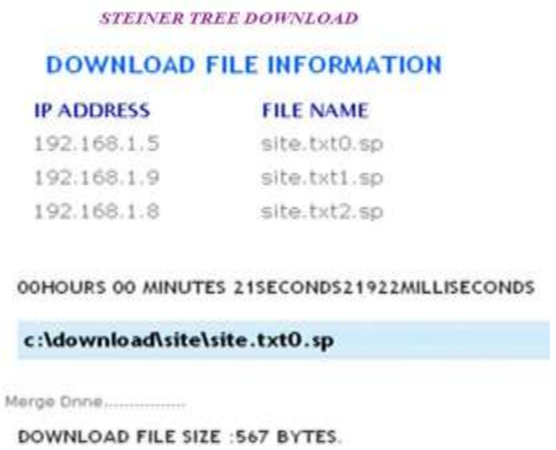
IP ADDRESS	FILE NAME
192.168.1.5	site.txt0.sp
192.168.1.9	site.txt1.sp
192.168.1.8	site.txt2.sp

00HOURS 00 MINUTES 22SECONDS22438MILLISECONDS

c:\download\site\site.txt0.sp

Merge Done.............

DOWNLOAD FILE SIZE :567 BYTES



STEINER TREE DOWNLOAD

DOWNLOAD FILE INFORMATION

IP ADDRESS	FILE NAME
192.168.1.5	site.txt0.sp
192.168.1.9	site.txt1.sp
192.168.1.8	site.txt2.sp

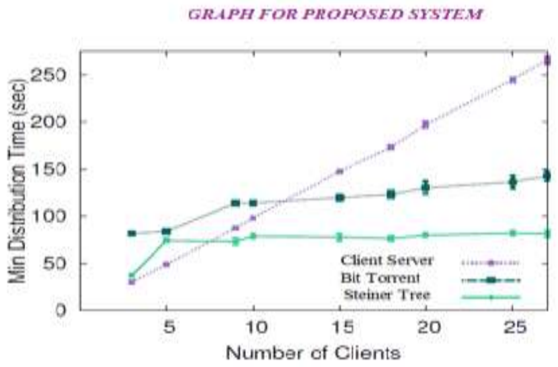
00HOURS 00 MINUTES 21SECONDS21922MILLISECONDS

c:\download\site\site.txt0.sp

Merge Done.................

DOWNLOAD FILE SIZE :567 BYTES.

Proposed System Graph:



FUTURE WORK

In our future work we use another content distribution algorithm named Steiner Tree Algorithm to reduce the upload and download time using a comparative work .The problem is to find a Steiner tree connecting the source node and destination nodes that minimizes data distribution time. To minimizing upload and download time the Algorithm implemented is BitTorrent and Steiner tree algorithm. Primarily the BitTorrent technology is applied in minimizing the Time and the required file is get uploaded. The uploaded file gets connected to the various clients to distribute the content which reduces the load of the file to get splited. Then the splited file from based on content distribution which reduces the load of the file to get splited. Then the splited file from various clients are get downloaded. The downloaded file will be viewed by displaying the Time limit by using the LAN connection and cloud formation through internet.The splited file can be seen as complete document by using the BitTorrent merging algorithm performance evaluated results that confirm the potential of our system.

CONCLUSION

The overarching goal of this setting is to get minimizing the maximum time of bulk synchronous content distribution where it takes any client in a set to download content of the required file. In the paper, we have developed a formulation of BitTorrent and Steiner Tree to split the content over the file, through the cloud resources in order to meet their MinimumDistribution objectives.

We presented performance evaluated results that confirm the potential of our system.

REFERENCES

[1] Raymond sweha, vatche Ishakian, Azer Bestavros, **Angels in the Cloud**, 2011 IEEE 4th International conference.

[2] B. Cohen, “Incentives Build Robustness in BitTorrent,” 2003.

[3] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and Venkataramani “Do incentives build robustness in BitTorrent,” in *NSDI*, 2007. International conference on cloud computing.

[4]Instrumented BitTorrent, https://gforge.inria.fr/projects/bt-instru/.

[5] J. Mundinger, R. Weber and G. Weiss, “Optimal Scheduling of peer-to peer File dissemination,” *Journal of Scheduling*, 2008.

[6] Harris, Jr., Frederick C. "Steiner Minimal Tree: Their Computational Past, Present, and Future." University of Nevada. http://www.cs.unr.edu/~fredh/papers/journal/smttcppaf/paper/paper.ps.gz.

Tunnermann2 summarized some of the points of criticism as follows: