International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Enhanced Distributed Downloading Technique using DTPX

Nithin SJ[#a], Manoj N[*b], Sahana MP[#c]

[#a] *Student, CSE, Dayananda Sagar College of Engineering, Bangalore, India.*
[#b] *Student, CSE, Dayananda Sagar College of Engineering, Bangalore, India.*
[#c] *Professor, CSE, Dayananda Sagar College of Engineering, Bangalore, India.*

---***---

Abstract— The utilization of Divide and Conquer Paradigm in the Distributed and Parallel Download technique has been analysed in this paper and the main focus is to test and explain how this new approach helps in the downloading of large files in fractional amount of time when compared to standard download which is present currently. To accomplish the task of downloading large files using a distributed divide and conquer paradigm, file is divided into multiple parts (download ranges) and the task of downloading each part is assigned to the peers available in the local network. Every participating peer then downloads the assigned file part and on downloading 5 megabyte of part assigned, each 5 megabyte data is packed as a data chunk and sent to the master client in real time. Hence, the complete file is made available on the master client by the time of completion of downloading all the parts and thereby overcoming the time required to transfer the parts separately once the part has been downloaded successfully. This protocol is named as Distributed Transfer Protocol Xtreme(DTPX).

Through this protocol, peak efficiency of the distributed downloading technique is achieved and the time taken by linear download is divided by n times, where n is the number of peers participating in the distributed download in a Distributed Bandwidth Local Area Network, or the user is connected to a Cellular Network.

Keywords—*Distributed Download, Divide and Conquer, Parallel computing.*

I. INTRODUCTION

Distributed computing is a means of breaking up a complex problem, distributing the pieces to various computers across a network (usually the Internet), and having them work on the problem concurrently until the problem is solved. The difference between distributed and parallel computing is that parallel computing is done on a single architecture with multiple processor cores and distributed computing is done over a network. According to Bill Godfrey, distributed computing is performed with a client-server model where the server sends out work packages and constructs the answer from the responses returned by the clients.

End users typically use serial download strategy to download files from the internet. To overcome the time consumption, Parallel Downloading technique emerged. Although, this technique is not used on a day to day basis due to lack of practicality in the application of the technique when small files are considered. This paper has two contributions : a) It removes the lack of practicality posed by the Distributed Download technique. b) It proposes an efficient approach towards the Distributed Download technique when the Download speed is kept as top priority and the bandwidth is distributed among the Local Area Network or the end user is connected to the Cellular Network.

The case where the user is connected to Cellular Network to download the file and the part retrieval is a further enhancement for now.

II. PREVIOUS WORK

The existing Parallel Downloading system requires the end users to be in the same network, and the downloading and merging is done one after the other. Although the existing Parallel Downloading system reduces the download time by 'n' times, where 'n' being the number of users which are parallely downloading the file, the splitted pieces are required to uploaded to the master client after finishing the download, which will consume time, considering the file is transferred through the 'FTP' in the local network, which has a typically ranges from 10-11 megabytes per sec.

Server with multiple mirror architecture has been proposed to overcome the load balancing problem and it has been proven effective for which it was intended to. Here the client can request various parts of a file which is hosted across the mirror servers on the basis of resource utilization and quality of the path.7

Parallel downloads has been implemented to some small extent in a P2P Downloading system called Torrents where files are split into hundreds, if not thousands of pieces and the user downloads random file pieces and uploads simultaneously, thereby distributing the load among the network and reducing the stress on the server side. If the goal is to reduce the time constraint, the

Hence there is a need to overcome the time consumption posed by the current system of distributed download using a more efficient approach towards Distributed Download.

III. Methodology

In this section, DTPX is discussed briefly. The proposed solution is to be implemented at the application level. There are four entities, i) Master Client, ii) Slave Client, iii) Main Server in a Client-Server Architecture and iv) Router

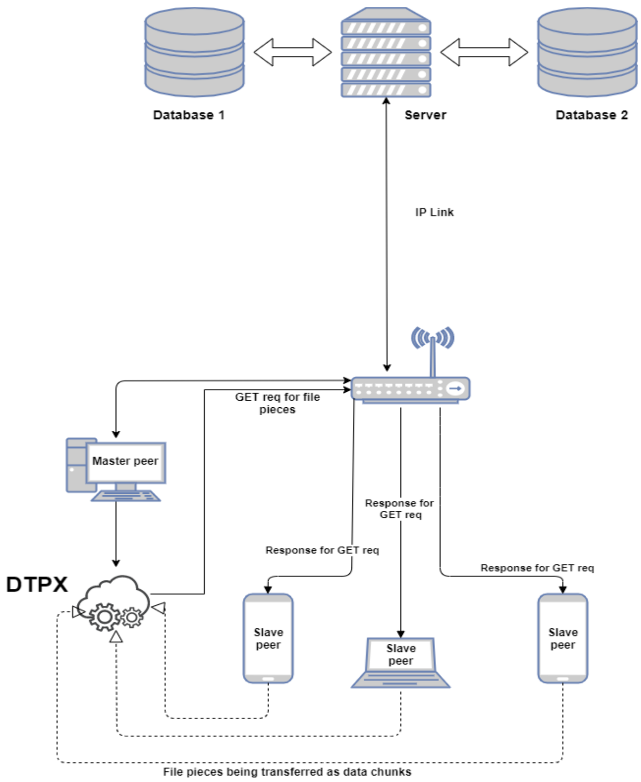
A. *Architecture*

1) *Master Client :* The Master client is responsible for the initiation of Distributed Download and allotting the certain parts which is to be downloaded by a specific slave client. The Master Client is also responsible for receiving parts of data from various slave clients and merging the data chunks such that the entire file is made available in the Master Server once every part is downloaded by the Slave Client.

2) *Slave Client :* The Slave Client is responsible for downloading the part allocated to it by the Master Client. It downloads the parts in a buffer of 5MB, encapsulates it in a Data packet, and sends to the Master Client.

3) *Server :* The Server delivers the file requested by the various clients by sending the response to the GET requests from the clients. The Server has access to the Database from where it fetches the file as requested by the client.

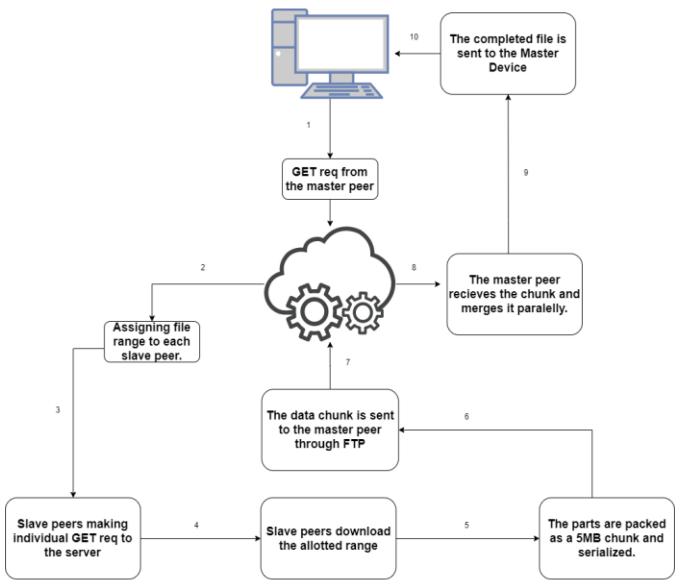
4) Router : Router provides Internet connection to all the connected clients. The bandwidth provided by the ISP is assumed to be distributed bandwidth, where all the clients get the same bandwidth and there is no sharing or splitting of the bandwidth upon load.



System overview of Distributed Download using DTPX

B. *Working*

The core principle of the protocol is the Divide and Conquer paradigm. Here the file to be downloaded is divided and distributed across various clients. The overview of the working is pictured below.



1) *Initiation*: The Master Client obtains the file URL and passes it to the DTPX application. Now the application decides the range of the file which is to be downloaded by the slave client.

2) *Request:* The Master Client assigns the range of 5 Megabytes to each of the Slave Clients in an orderly manner. The application running on the slave client, sends an HTTP GET request to the server for a specified file range. This HTTP request has an extra header called 'Range', containing the range of a file which is to be downloaded. The server responds back with a '200 OK status code' and 'Accept-Range : bytes' indicating that request is good to go. If the range specified is illegal, it responds with a '200 status code' with 'Accept-Range : none'. If the specified range is not supported or if it cont

3) *Download:* The assigned parts of the file are downloaded simultaneously in the Slave client as well as the Master client and for each 5 Megabyte of data downloaded, the Slave client creates a data chunk and sends it the DPTX application. This data chunk is in the format of JavaScript Object Notation(JSON) and has a property of 'count' which indicates the order of the chunk.

4) *Merge*: DTPX application in the Master client merges the received data chunks simultaneously while downloading the remaining parts of the file, such that when tha parts are downloaded, the Master has the file in its completed state.

5) Verify: If the website has an MD5 checksum provided with the file, the DTPX application provides a means to verify the integrity of the file through MD5 Checksum verification. This step is optional, and it is used during the testing phase of the application.

C. Scenarios

1) The Slave Client is Disconnected during the Download

If the Slave client is disconnected abruptly. The DTPX application has a timer for each data chunk to be retrieved, say 30 seconds. If the Master fails to receive the 'next count' after 30 secs, the slave is declared as Disconnected, and the range is passed onto another available client and the process continues.

2) Merging process takes more time than receiving of parts

The parts which are received are stored separately in a pool. Since the chunks have count property, the merging process is exclusive.

D. Expected Results

1) Through Regular Linear Download,

The total time taken to download a 1GB file through 1MBPS connection,

T = (P)/(S)

where P = File size in Megabytes,

S is the Connection speed in Megabytes,

T is in seconds.

T = (1024MB) / (1MBPS) = 1024 seconds

= approx. 17 mins.

2) Through Current Distributed Download,

To download the file pieces

(1024MB) / 2 = 500 MB = 8.5 minutes .. (i)

To transfer the file pieces to the master.

(500MB) / (10MBPS) = 50 secs .. (ii)

Here the 10MBPS is the assumed speed of an FTP transfer in a regular Local Area Network.

Total Time consumed = 9.4 minutes. (50 - 20 = 30% efficient when compared to Regular Linear Download)

3) Through Distributed Download using DTPX

To download the file pieces

(1024MB) / 2 = 500 MB = 8.5 minutes (50% efficient)

The extra time required to merge the received pieces is 0.

Hence,

8.5 minutes + 0 minutes = 8.5 minutes

50% efficient than the Regular Linear Download and 20% more efficient than the existing Distributed Download

E. Possible Applications

1) *Content Servers containing huge media files.*

In production environments, oftentimes huge RAW media files are hosted on a Local server, and the users who are wanting to download the files have to wait for FTP transfer to complete. This wait can be reduced significantly through the DTPX protocol.

2) *To overcome the FUP Limit restrictions posed on individual IP addresses*

Users can utilize the bandwidth available from other clients as well.

3) *Utilization of Multiple Network Interface*

Users can remove the external device involvement, if the client has multiple Network Interfaces configured individually and can perform the Distributed Download.

IV. CONCLUSIONS

The version of the Distributed Download promises efficiency in reducing the Download Time significantly and various applications of this protocol are not yet explored. A software client, preferably with a cross platform functionality, which can implement this protocol is to be developed. The existing multipart download clients can implement this protocol to take advantage of local clients bandwidth and increase its efficiency.

ACKNOWLEDGMENT

I'd like to thank Prof. Sahana MP from the Computer Science Department of Dayananda Sagar College of Engineering for her guidance and advice. I'd also like to thank our college for providing amazing student-teacher interaction which taught me many things.

REFERENCES

[1] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.

[5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[8] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.

[9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.