Parallelized FTP

Shaleeza Sohail, Sanjay Jha and Hossam ElGindy
School of Computer Science and Engineering
The University of New South Wales, Sydney 2052, Australia
Email: {sohails,sjha,elgindyh}@cse.unsw.edu.au

UNSW-CSE-TR-0301

March 2003

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY · AUSTRALIA

Abstract

The parallelized FTP (P-FTP) approach, attempts to solve the problem of slow downloads of large multimedia files while optimizing the utilization of mirror servers. The approach presented in this paper downloads a single file from multiple mirror servers simultaneously, where each mirror server transfers a portion of the file. The P-FTP server calculates the optimum division of the file for effecient transfer. The dynamic monitoring ability of P-FTP maintains the file transfer process at the optimized level no matter how abruptly network and mirror server characteristics change.

1 Introduction

Downloading large multimedia files on the Internet is a very time-consuming, slow process. Provision of multiple mirror servers and efficient compression mechanisms are two of the approaches that try to address this problem. The most reliable and widely used protocol for file transfer is File Transfer Protocol (FTP). The files are usually replicated on multiple mirror servers and the client tries to download file with FTP, from the server which is geographically closest to it. Selection of geographically closest server is based on the assumptions that, firstly, the transfer will create least congestion on the Internet and secondly, the file is transferred in minimum possible time. However, it is possible that the closest server is the one which is most highly utilized and the links between that server and the client are highly congested so the optimised download process needs other criteria for selection of best mirror server. Number of research efforts have addressed this problem of selecting the best server for a particular client on the basis of different metrics. Guyton and Schwartz has proposed a server selection technique in [1] on the basis of hop counts and round trip delay. Carter and Crovella have introduced congestion, measurement of latency and speed of bottleneck link in the metrics used in server selection technique [2], however they have not considered server's utilization. Fu and Venkatasubramanian took the complexity of the server selection technique one step further by introducing the server's availability in terms of available CPU cycles, I/O bandwidth and memory in addition to the characteristics of the path i.e; delay and available bandwidth [3]. All these approaches address the problem of selecting the best single server to download data. Buyers et al have presented the idea of downloading from multiple servers and peers simultaneously in peer to peer scenario but they failed to consider the server's availability and network QoS issues [4]. Few peer to peer applications like Kazaa [9] and Furthurnet [10] support simultaneous download of a single file from multiple peers. All peer to peer applications allow direct download from the peers who has the copy of the desired file. The peer to peer approach tries to facilitate file sharing among users without a centralized server. Due to this common property of these approaches the file transfer process is random and unoptimized. Moreover approaches used in [4] [9] [10] does not support a central entity to monitor and dynamically optimize the download process with respect to a desired QoS parameter due to the distributed nature of peer to peer network.

This paper proposes an approach to optimize the process of downloading a file with FTP by selecting multiple servers on the basis of server availability and path quality. Simultaneous download from multiple servers decreases the delay for the client and reduces the burden on any single server and path. The server availability can be defined in terms of utilization of CPU, memory and I/O bandwidth of that server at that time, however more attributes can be added. The path quality depends upon the availability and QoS provided on that path at that instance. Using these values the servers are checked for suitability to download data. The amount of data downloaded from any server depends upon its suitability. The P-FTP ap-

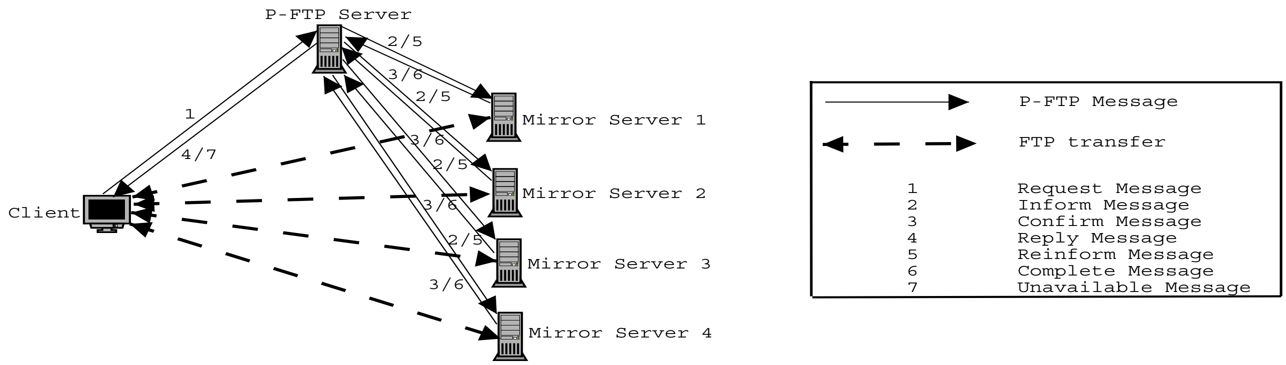

Figure 1: Interaction Among P-FTP Entities

proach introduces a central entity, P-FTP server, that dynamically monitors the active P-FTP sessions and recalculates the mirror servers suitability in case there is a change in path quality, mirror server utilization and availability. The self monitoring functionality of P-FTP server forbids itself to become a bottleneck in network by refusing new requests after it reaches its maximum utilization.

The rest of the paper is organized as follows. Section 2 briefly explains P-FTP algorithm and functionality of its components. Section 3 describes the simulations that were performed to study the P-FTP algorithm. Section 4 concludes the paper and identifies few future work options.

2 Parallelized FTP (P-FTP)

In this section the P-FTP approach is discussed briefly. The P-FTP service is to be implemented at application level. The three main entities in the P-FTP application are the requesting client, the P-FTP server and the mirror servers. The P-FTP server controls the overall functionality of the application and is to be placed in the Internet in such a strategical manner by the P-FTP service provider that its communication does not effect any other service on the Internet. Whenever a client wants to download a file, it sends a request message to the P-FTP server. On receiving the request, the P-FTP server collects information from its database about the mirror servers that contain the copy of the requested file. The P-FTP server runs the suitability algorithm to calculate and evaluate the suitability of the mirror servers. The suitability algorithm is discussed in the section 2.1. Respective suitability value indicates the portion of file that is to be downloaded from each mirror server. The P-FTP server sends an inform message to all mirror servers, which contains the suitability of that mirror server, the requested file and requesting client information. On receiving the inform message mirror server stores the information and sends confirm message to the P-FTP server. The P-FTP server sends the reply message to the requesting client, which contains mirror servers information and their respective suitability. The client initiates the FTP session with all mirror servers. On receiving the FTP request from the client, the mirror servers look at the stored information. If a related inform message was received previously then the portion of file mentioned in the inform message is transferred to the client. If no inform message was received, the mirror server treats the request as a traditional FTP request and transfers the complete file. From the reply message sent by P-FTP server, client knows the portions of file that each mirror server is transferring. The client uses that information to reassemble the file after the file transfer from each mirror server is completed. After the completion of transfer the mirror servers send complete message to the P-FTP server, which triggers the removal of that request from the active sessions. When the P-FTP server receives database update, the suitability values for all active P-FTP sessions are recalculated and mirror servers are notified accordingly with reinform messages. The mirror servers update the file portion values for the related active and inactive P-FTP sessions according to the reinform message. A new reply is sent to the client to inform new suitability values. Figure 1 shows the interaction among the P-FTP entities.

2.1 The Suitability Algorithm

The core of the P-FTP application is the suitability algorithm. The main function of the suitability algorithm is to calculate the suitability of mirror servers on the basis of the optimization policy. After receiving the request from the client, the P-FTP server runs the suitability algorithm and optimum suitability of the mirror servers is calculated. The suitability of each mirror server indicates the portion of requested file that should be transferred from that server. However it is possible that due to the utilization of any mirror server at the time of request, the transfer of calculated file portion from that mirror server is not possible. In that case the algorithm runs recursively till the optimized suitability of all mirror servers is calculated while keeping the utilization of mirror servers under maximum threshold value.

The complete P-FTP application at the P-FTP server can be step-wise explained by algorithm ***P-FTP***. Suppose the client requests for file X, the P-FTP server finds the resources required to download that file, FR and the set of mirror servers, M that have replicated copy of that file.

$$M \subset MS \tag{1}$$

Where MS is a set of mirror servers, that are registered with P-FTP server. The suitability algorithm finds the suitability S_m for all members of $m \varepsilon M$, on the basis of the optimization policy. The suitability of all mirror servers is checked against their available resources, AR. If check fails for any mirror server, the suitability of that mirror server is reduced so that its utilization remains less then the maximum threshold value and that mirror server is replaced from M to M_{Final} set.

$$M_{Final} \subset MS \tag{2}$$
$$M_{Final} \subseteq M \tag{3}$$

Where M_{Final} is the set of mirror servers to whom P-FTP server will send inform message after the suitability algorithm finishes.

Algorithm *P-FTP*

```
Initialize M_Final
Find FR of X
Find M for X
PFTP-algo( M ) [
if M_Final = ∅
  then if M ≠ ∅
        then Calculate S_m, ∀m using optimization policy
             Calculate A_m ← S_m * FR, ∀m
             if ∀m, AR_m > A_m
               then Add all m to M_Temp
                    Delete all m from M
                    Add All m_Temp to M_Final
               else ∀k, S_k ← AR_k/FR
                    s.t.k ∈ [1, 2, ....m]
                    FR ← (FR − (S_k * AR_k))
                    Add k to M_Temp
                    Delete k from M
             PFTP-algo( M )
        else Send resources unavailable message to client
  else Send Inform message to m_Final
      if confirm message received from all m_Final
        then Send reply message to Client
             Delete all m_Final from M_Final
        else Copy all m_Final from M_Final to M
             Delete all m_Final from M_Final
             Delete not responding m from M
             PFTP-algo( M )
]
```

2.2 P-FTP Server Monitoring

The mirror servers send alive messages to P-FTP server frequently for monitoring purpose. If the P-FTP server does not receive three consecutive alive messages from any mirror server, that mirror server is considered to be down. The down mirror server is not included in any new suitability calculations, however if that mirror server is part of any active P-FTP session then P-FTP server recalculates the suitability for that active session without considering down server and send reinform messages to all other mirror servers. If all mirror servers participating in an active P-FTP session are down, a reply message is sent to the requesting client to inform the failure of the request. The P-FTP server needs two consecutive alive messages from a down mirror server to change its status to up.

The P-FTP server has to monitor the mirror servers as well as the active requests, that is why each request contributes significantly to the P-FTP server processing burden. Before accepting a new request, the P-FTP server considers its own utilization. In case P-FTP server finds its own utilization near maximum threshold value, it responds the requesting client with unavailable message. At the client side a timeout mechanism is introduced. The client sends the request for the same file to the P-FTP server at most three times after random interval of times. If client receives three unavailable messages from the P-FTP server, it starts downloading the file using traditional FTP approach from single mirror server. The self monitoring ability of the P-FTP server prevents the creation of processing bottleneck at the P-FTP server, which can degrade the overall network performance.

2.3 Database

P-FTP server keeps a comprehensive database about mirror servers and network. The mirror server information consists of load conditions and the replica map of stored files. The network information includes network topology and values for different network parameters. The mirror servers register them self with P-FTP server at the time of initialization. These servers provide information to P-FTP server about their own utilization and replicated files. The soft state information about mirror servers is updated constantly by the mirror servers. The network information which needs constant update consists of QoS parameters along multiple paths between networks and the utilization of network links.

Numerous researchers are trying to design a tool that can accurately measure network parameters between arbitrary Internet end hosts while producing least burden on network. The basic traditional approach to measure latency in the Internet is with tools like Ping and Traceroute. The use of these tools is easy but the measurements cannot be highly accurate. Sting is a tool that uses TCP protocol to measure the network attributes [5]. IDMaps is designed as an underlying service to provides the distance information, however IDMaps requires deployment of additional infrastructure [6]. King is a tool that estimates the latency between arbitrary end hosts with the help of existing DNS infrastructure [7]. Eugene Ng et al proposes the measurement of transmission delay between the peers in peer-to-peer architecture with the help of coordinates-based mechanism [8].

The P-FTP approach may use any of the above mentioned approaches to accurately calculate network parameters. However, the frequency at which these measurements are taken is a very important factor in the efficiency of P-FTP. By setting the frequency value too low the efficiency of P-FTP algorithm can be reduced considerably and setting the value unnecessarily high can increase network load exponentially. Due to the dynamic nature of Internet there exists a direct relation between the accuracy of these measurements to the frequency at which the measurements are made. However, the direct relation between the overhead produced by such measurements to the measurement accuracy greatly effects measurement frequency. The nature and size of the network topology greatly influences these two relations. To optimize P-FTP benefits more investigation is required in order to fix these two relations. A major part of our future intended work is the study of these relations in different topologies.

The P-FTP benefits are greatly dependent upon the accuracy of its database. However, the large number of database updates can flood the network and intern reduce the overall network performance. Gathercast [11] and concast [12] are multipoint to point packet gathering approaches that try to reduce the network traffic by merging small packets traveling towards a common destination. These approaches try to reduce packet processing at the routers by merging number of small packets into one big packet which can reduce the packet loss rate at routers and hence improve the network performance. Introducing gathercast and concast services in P-FTP approach to combine, FTP packets from multiple servers to same client and the database update packets destined to the P-FTP server can increase the network performance and P-FTP efficiency.

2.4 Optimization Policy

The optimization policy defines one or more characteristics of mirror server on the basis of which their suitability is calculated, that characteristic of the mirror server is called the Optimization Variable (OV) and is represented as V. The algorithm for the calculation of suitability tries to optimize the process of file download with respect to OV. The optimization policies can be categorized on the basis of number of OV. The two categories of optimization policies for P-FTP are discussed briefly in the following subsections.

2.4.1 Single Variable Optimization Policy

Single variable optimization policy defines one optimization variable. If there are m mirror server that has the replicated copy of requested file then OV values, V_k for all m are collected from the database. The method to calculate the suitability of mirror servers depends upon the type of OV. The OV can be of two types, direct OV and inverse OV. Direct OV values directly influence the suitability of mirror servers. The example of direct OVs are bandwidth along the path and mirror server's available resources. The suitability, S_k of mirror servers depending upon the direct OV values is calculated by the following equation:

$$S_k = \frac{V_k}{\sum_{k=1}^{m} V_k} \tag{4}$$

Inverse OV values inversely effect the suitability of mirror server. End-to-end delay along a path and mirror server utilization are the examples of inverse OV. In the case of inverse OV, the minimum OV value, V_z is searched and S_k values are calculated by the following equations:

$$s_k = \frac{V_z}{V_k} \tag{5}$$

$$S_k = \frac{s_k}{\sum_{k=1}^{m} s_k} \quad (6)$$

Two of the most obvious and important single variable optimization policies are lowest delay policy and best utilization policy. In Lowest Delay Policy (LDP) end-to-end delay between mirror servers and the requesting client is the optimization variable. LDP tries to download file in minimum possible time. The optimization variable in Best Utilization Policy (BUP) is the utilization of mirror servers. Due to the highly unreliable and unpredictable nature of Internet, it is possible that QoS is not the primary focus for the parallelized FTP transfers, in that case BUP is considered which tries to keep the utilization of all mirror servers at minimum.

2.4.2 Multiple Variable Optimization Policy

The importance of different characteristics of mirror servers can vary with respect to time of day or network characteristics. Multiple variable optimization policy can cater for these requirements. One example of such policy is delay and utilization policy.

Delay and Utilization Policy (DUP) is a double variable optimization policy which has optimization variables of end-to-end delay and the server utilization. The suitability of mirror servers is firstly calculated using LDP and BUP. The final suitability, S_F is calculated as:

$$S_F = S_{LDP} * a + S_{BUP} * b \quad (7)$$

$$a + b = 1 \quad (8)$$

S_{LDP} and S_{BUP} are calculated suitability values using LDP and BUP policies respectively. a and b are parameters whose values depend upon the relative importance given to each policy.

3 Simulation

A simulation is performed using discrete event simulator specially programmed for networking research, Network Simulator 2 (NS-2). A new application and a new agent is introduced to test the idea of multiple FTP transfers to a single client simultaneously. The simulation results are validated using quantitative stochastic simulator Akaroa-2 with NS-2.

3.1 Topology

The network topology is created by Boston University Representative Internet Topology Generator (BRITE), however, to make the topology more realistic some minor modifications were made. The topology is shown in Figure 2, it has 10 nodes and 19 links. Each node in Figure 2 is a router and there are multiple paths between all routers. Ten mirror servers and 5 requesting clients are placed directly behind

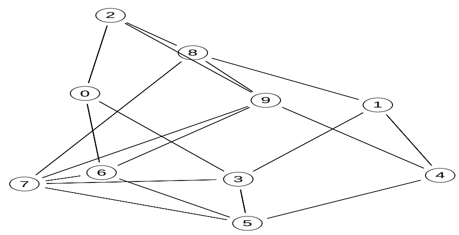

Figure 2: Topology

Mirror Server	Delay(sec)	Suitability
3	0.03024	0.26915
4	0.0452311	0.179945
5	0.02016	0.403725
6	0.0553004	0.14718

Table 1: Mirror Servers Delay and Suitability

routers randomly and are not shown in Figure. Multiple traffic sources and sinks are introduced in the network randomly to produce the contesting traffic.

3.2 Simulation Methodology

In simulation, the client requests a file randomly from P-FTP server, file size ranges from 500 KByte to 5 MByte. Each file is replicated on at most five mirror servers. The P-FTP server calculates the mirror servers suitability and indicates the result to the mirror servers and the requesting client with inform and reply messages respectively. The mirror servers start transferring the portion of file to the client after receiving a FTP request from client. On receiving complete file the client calculates the time taken to complete the request. The time is calculated from the instance when the client first requested the P-FTP server to the instance file transfer was completed, the values are averaged over multiple runs. The mirror servers utilization is updated by the mirror servers before the start of transfer of file and after the file transfer is finished. The end-to-end delay from each mirror server and each client is calculated by monitoring agents introduced on the servers and clients. The end-to-end delay values are updated after every 200 seconds.

3.3 Simulation Results

The effect of P-FTP on download time is explained by comparing the file transfer process of P-FTP to simple FTP by considering an example of client 5 and file 4. File 4 is replicated on four mirror servers 3,4,5 and 6. The end-to-end delay values between client 5 and mirror servers taken from the database and mirror servers suitability values calculated by P-FTP server using LDP are shown in table 1. Mir-

Mirror Server	Download Time (sec)
3,4,5,6 (P-FTP)	2.3163
3 (FTP)	6.23067
4 (FTP)	9.33167
5 (FTP)	4.15397
6 (FTP)	11.4014

Table 2: Download Time from Single Server

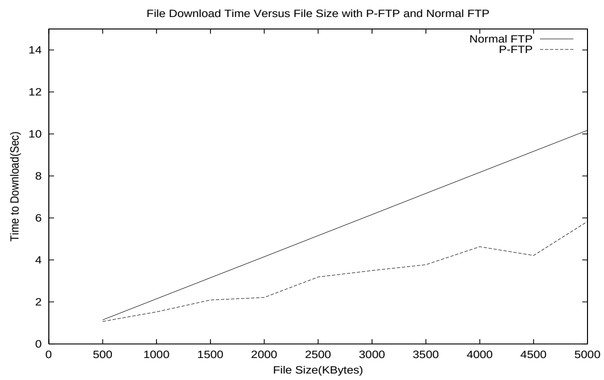

Figure 3: Simulation Results

ror servers 3,4,5, and 6 transfer the portion of file 4 to client 5 according to their respective suitabilities. Client 5 calculates the download delay after receiving complete file using P-FTP approach. Client 5 downloads file 4 from each mirror server separately using traditional FTP approach and calculates the respective download delays. Table 2 shows the download time for P-FTP and traditional FTP approach.

Figure 3 shows improvement in terms of time when files of different sizes are downloaded using P-FTP approach as compared to traditional FTP approach. The time difference is more obvious for large files then small files as in later case the overhead of the message exchange between the P-FTP entities overshadow the delay measurement. The time measurements in Figure 3 with P-FTP approach does not follow a straight line, however the download time for normal FTP is directly proportional to the file size. The reason is that for P-FTP the files are being downloaded from different set of servers however for normal FTP all the files are downloaded from same mirror server, which is geographically closest to the requesting client.

4 Conclusion

The paper presents a parallelized FTP approach which tries to solve the slow download problem of large multimedia files on the Internet. Simultaneous download of a single file from multiple mirror servers optimizes the download process with respect to defined network and/or server characteristic. Different optimization policies provide mechanism to enforce particular criteria for selection of appropriate mirror servers to download file. The P-FTP monitoring ability dynamically updates the division of file during file transfer to keep the file transferring process at optimum level.

The simulation results show significant improvement in download time for large files when P-FTP approach is used as compared to traditional FTP, however much work remains to be done. The study of database update frequency effect on download time in P-FTP approach is our next goal. The impact of different approaches used to calculate the network parameters is to be studied in detail in future. The introduction of P-FTP mechanism in video on demand and video streaming scenarios needs more investigation. To monitor the effect of integration of concast and gathercast approaches with P-FTP is also part of our future work.

References

[1] J. Guyton and M. Schwartz, *Locating Nearby Copies of Replicated Internet Servers*, In Proceedings of SIGCOMM 1995, pages 288-298.

[2] R. Carter and M. Crovella, *Dynamic server selection using bandwidth probing in wide area networks*, In Proceedings of INFOCOM 1997.

[3] Z. Fu and N. Venkatasubramanian, *Directory based composite routing and scheduling policies for dynamic multimedia environment*, In Proceedings of IEEE International Parallel and Distributed Processing Symposium 2001 (IPDPS 2001).

[4] J. Byers et al, *Informed Content Delivery Across Adaptive Overlay Networks*, In Proceedings of ACM SIGCOMM Conference 2002, Pittsburgh, PA.

[5] S. Savage, *Sting: a tcp-based network measurement tool*, In Proceedings of the 1999 USENIX Symposium on Internet Technologies and Systems (USITS '99), October 1999.

[6] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, *IDMAPS: a global Internet host distance estimation service*, In Proceedings of IEEE/ACM Trans. on Networking, October 2001.

[7] K. Gummadi, S. Saroiu, S. Gribble, *King estimating Latency between Arbitrary Internet End Hosts*, In Proceedings of the SIGCOMM Internet Measurement Workshop (IMW 2002).

[8] E. Ng and H. Zhang, *Predicting Internet network distance with coordinates-based approaches*, In Proceedings of IEEE INFOCOM 2002, New York, NY, June 2002.

[9] www.kazaa.com

[10] www.furthurnet.com

[11] B. Badrinath and P. Sudame, *Gathercast: the design and implementation of a programmable aggregation mechanism for the internet*, In Proceedings of ICNP, 1998.

[12] K. Calvert, J. Griffioen, A. Sehgal and S. Wen, *Building a programmable multiplexing service using concast*, In Proceedings of International Conference on Network Protocols, 2000.