A Comparison of Efficiency of Data Transfer by Using Rsync, Rsync+SSH and Dropbox

Wiman Banditvilai and Sirapat Boonkrong

Abstract—**This paper presents a comparison of data transfer by using three protocols: Rsync, Rsync+SSH and Dropbox on Ubuntu. The efficiency or the speed of the data transfer is measured through data downloads and data retrievals. The results show that Rysnc is the most efficient of the three protocols in both transfer speed and bandwidth usage.**

Index Terms—**Rsync, Rsync+SSH, Dropbox.**

I. Introduction

Technology and the Internet are becoming popular. They are widely used in homes and offices through local area networks or wireless local area networks. Data is transferred to client from client (peer to peer) and server to client. Users can connect to networks every place with the internet and data transfer can be done by many protocols, such as HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), Rsync and others.

Data transfer can be completed in different ways depending on individual's requirements. e.g. to download data, to watch a movie or to listen to music. Data transfer will have similar characteristics but the format of each protocol is often different. Parallel File Transfer Protocol (P-FTP) is designed to move large data from one cluster to another as fast as possible and introduces the concept of data transfer using multiple parallel data paths between clusters [1]. It attempts to solve the problem of slow downloads of large multimedia files [2].

TCP is optimized for accurate delivery rather than timely delivery [3]. In addition, one must consider the security of the computer and the Internet networks [4]. The SSH File Transfer Protocol (also Secure File Transfer Protocol, or SFTP) is a network protocol that provides file access, file transfer, and file management functionalities over any reliable data stream. It was designed by the Internet Engineering Task Force (IETF) as an extension of the Secure Shell protocol (SSH) version 2.0 to provide secure file transfer capability, but is also intended to be usable with other protocol [3].

Rsync is a free computer software for Unix and Linux which synchronizes files and directories from one location to another while minimizing data transfer using delta encoding [5]. Rsync is a protocol for data transfer and streaming media that uses a single round of messages between two machines [6] and transmits only changed data from the source to the destination to save bandwidth usage [7] as it uses compression and decompression method while sending and receiving data both ends [8]. Rsync has established itself as a reliable backup tool in small and medium scale businesses [9]. It is because small businesses need data backup tool as we know where pricing and reliability are important.

Dropbox is an easy-to-use, secure and efficient solution for sharing information with users across the internet. It allows users to upload files and specify recipient e-mail address from a web browser [10]. Dropbox reduces the amount of exchanged data by using delta encoding when transmitting chunks. The Dropbox service performance is highly impacted by the distance between the clients and the data-centers. The usage of per-chunk acknowledgements in the client protocol combined with the typically small chunk sizes deeply limits the effective throughput of the service [11].

The purpose of this study is to compare the efficiency of data transfer by using Rsync, Rsync+SSH and Dropbox.

II. Related Work

A lot of work has been done to study single-round synchronization techniques that achieve savings in bandwidth consumption while preserving many of the advantages of the Rsync approach. The resulting protocols give significant benefits over Rsync particularly on data sets with high degrees of redundancy between the version [12]. Tridgell, A. and Mackerras, P. also conducted research on the Rsync algorithm and found that single-round protocols [5],[6] are preferable in scenarios involving small files and large network latencies (e.g. web access over slow links). The best-known single-round protocol is the algorithm used in the widely used Rsync open-source tool for file synchronization of file system across machines [5]. In 2005 Mihaylov Irmak and T. Suel study single-round protocol for file synchronization that offer significant improvements over Rsync and reveal that a single-round protocol is feasible and communication-efficient with respect to a common file distance measure, and another protocol that show promising improvements over Rsync in experiences [6]. Deepak Gupta and Kalpana Sagar discuss the remote file synchronization protocols and compare the performance of all these protocols on different data sets [13]. The experiments have shown that multi-round protocols can provide significant bandwidth saving over single-round protocols on typical data set.

Expanding upon this single-file transfer research, this paper tests the efficiency of single-file and multi-file

Manuscript received October 19, 2013; revised January 9, 2014.

Wiman Banditvilai and Sirapat Boonkrong are with King Mongkut's University of Technology North Bangkok, Bangsue, Bangkok 10800, Thailand (e-mail: wb-ping@hotmail.com, sirapatb@kmutnb.ac.th).

DOI: 10.7763/IJCTE.2014.V6.861

transfers using Rsync, Rsync+SSH and Dropbox.

III. Experimental Setup

The equipment was set up as shown in Fig. 1 for testing data transfer by installing Ubuntu operating system and Rsync, Wireshark and Dropbox in both server (com A) and client (com B). Server (com A) ran Rsync server standby by using a wireless network connected to the Internet and transferring data to client (com B).

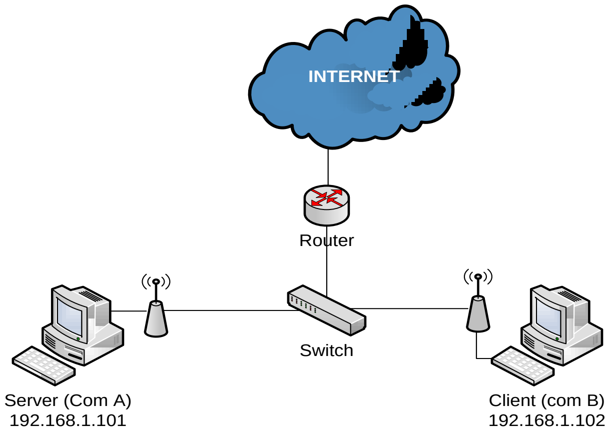

Fig. 1. The process of data transfer.

IV. Data Processing

To test the efficiency of data transfers by using Rsync, Rsync+SSH and Dropbox on Ubuntu, the average time spent for each protocol was used to compare the efficiency of data transfer. Each time was done in a sequence of Rsync, the Rsynce+SSH and then Dropbox. In this manner the different protocols were tested under the same networks conditions each time. The 100 test times were at various different periods of the day and week in order to find the efficiency under widely varying network conditions.

A. Test System by Sending Individual Files of 1MB, 5 MB and 10 MB

- To test the average time of data transfer protocol by sending individual files of: 1MB, 5 MB and 10 MB. It was tested 100 times to find the average time spent.

B. Test System by Sending 2 Files at a Time

- To test the average time of data transfer protocol by sending 2 files of: 1MB and 5 MB concurrently. It was tested 100 times to find the average time spent.
- To test the average time of data transfer protocol by sending 2 files of: 1 MB and 5MB concurrently where the Client (com B) already has a 1MB file. It was tested 100 times to find the average time spent.

C. Test System by Sending 3 Files at a Time

- To test the average time of data transfer protocol by sending 3 files of: 1 MB, 5MB and 10 MB concurrently. It was tested 100 times to find the average time spent.
- To test the average time of data transfer protocol by sending 3 files of: 1 MB, 5 MB and 10 MB where the Client (com B) already has a 1 MB and 5 MB files. It was tested 100 times to find the average time spent.

V. Experimental Results

A. Test System by Sending Individual Files

TABLE I: The Average Time of Data Transfer by Sending Individual Files of: 1 MB, 5 MB and 10 MB Using Rsync, Rsync+SSH and Dropbox

Scenario	Rsync	Rsync + SSH	Dropbox
1 MB	0.386 sec.	0.665 sec.	5.226 sec.
5 MB	0.881 sec.	1.185 sec.	5.474 sec.
10 MB	1.265 sec	1.554 sec.	5.749 sec.

In Table I, data transfer of sending individual files of: 1 MB, 5 MB, and 10 MB showed that Rsync spent the least time for data transfer at 0.386, 0.881, and 1.265 sec. respectively, while Rsync +SSH spent 0.665, 1.185, and 5.474 sec. and Dropbox spent the longest time of 5.226, 5.474, and 5.749 sec.

The results of the average time of data transfer by sending individual files are presented in Fig. 2.

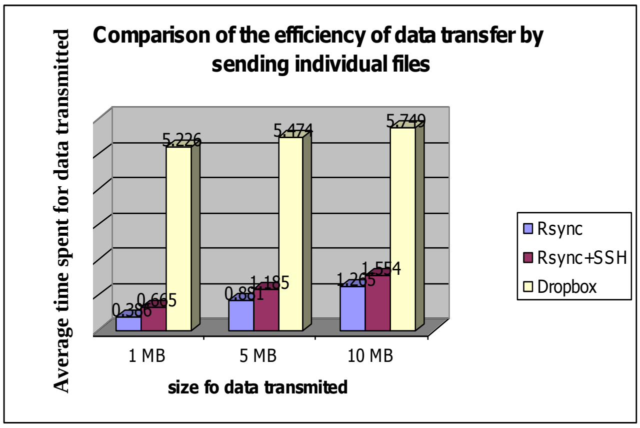

Fig. 2. Comparison of the efficiency of data transfer for sending individual files.

Fig. 2 shows that Dropbox spent the longest time for data transfer. This is because each data transfer had to be sent to the Cloud. Cloud is networks which abstract processing tasks [14]. For data transfer using Rsync+SSH, it was directly sent to client from the server so it took less time than Dropbox.

B. Test System by Sending 2 Files

TABLE II: The Average Time of Data Transfer by Sending 2 Files: 1 MB and 5 MB Concurrently Using Rsync, Rsync+SSH and Dropbox

Scenario	Rsync	Rsync + SSH	Dropbox
1 MB and 5 MB	0.946 sec.	1.209 sec.	4.994 sec.
1 MB and 5 MB (client side already has 1 MB file)	0.539 sec.	0.934 sec.	4.581 sec.

In Table II, the average time of data transfer of 2 files: 1 MB and 5 MB concurrently shows that Rsync spent the least time for data transfer at 0.946 sec. while Rsync +SSH spent 1.209 sec. and Dropbox spent the longest time of 4.994 sec. The average time of data transfer using Rsync, Rsync +SSH and Dropbox by sending 2 files of: 1 MB and 5 MB concurrently where the destination client already had a 1 MB file shows that Rsync spent the least time at 0.539 sec. while Rsync+SSH spent 0.934 sec. and Dropbox spent the longest time of 4.581 sec.

The results of the average time of data transfer by sending 2 files concurrently are presented in Fig. 3.

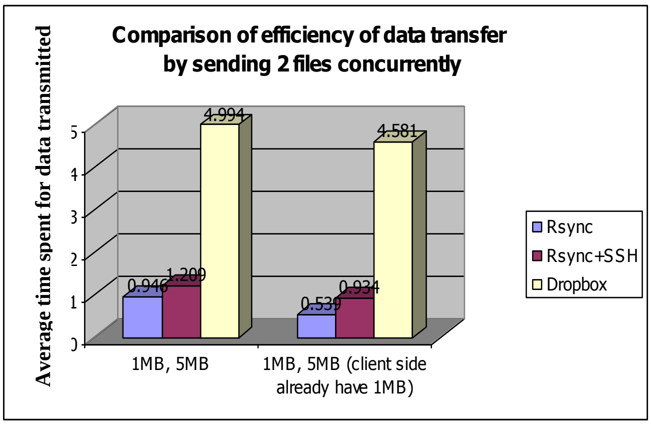

Fig. 3. Comparison of the efficiency of data transfer for 2 files concurrently.

Fig. 3 shows that the average time of data transfer using Rsync spent less time than through Rsync+SSH and Dropbox for sending 2 files concurrently: 1MB and 5 MB. Test System by Sending 3 Files

TABLE III: THE AVERAGE TIME OF DATA TRANSFER BY SENDING 3 FILES: 1 MB, 5 MB AND 10 MB CONCURRENTLY USING RSYNC, RSYNC+SSH AND DROPBOX

Scenario	Rsync	Rsync + SSH	Dropbox
1MB, 5MB, and 10 MB	1.998 sec.	2.171 sec.	5.416 sec.
1MB, 5MB, and 10 MB (client side already has 1 MB, and 5 MB file)	1.345 sec.	1.547 sec.	4.646 sec.

In Table III, the average time of data transfer of 3 files: 1 MB, 5MB, and 10 MB concurrently shows that Rsync spent the least time for data transfer at 1.998 sec. while Rsync+SSH spent 2.171 sec. and Dropbox spent the longest time at 5.416 sec. The average time of data transfer using Rsync, Rsync+SSH and Dropbox by sending 3 files: 1 MB, 5MB, and 10 MB concurrently where the destination client already has 1 MB file, and 5 MB files. The results show that Rsync spent the least time at 1.345 sec. while Rsync+SSH spent 1.547 sec. and Dropbox spent the longest time at 4.646 sec.

The results of data transfer from Table III are presented in Fig. 4.

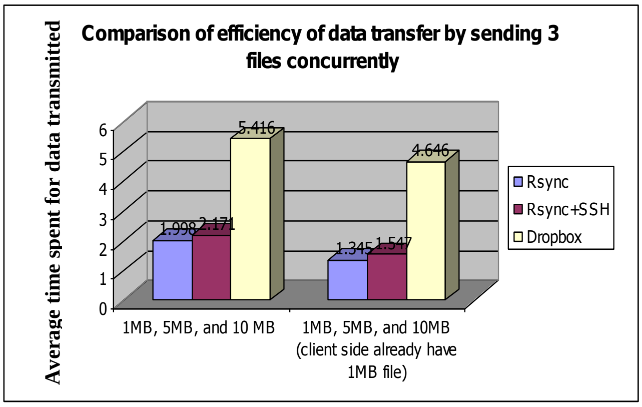

Fig. 4 Comparison of the average time of data transfer when sending 3 files concurrently.

Fig. 4 shows that data transfer using Rsync spent less time than Rsync+SSH and Dropbox.

VI. CONCLUSION

The result of comparing the efficiency of data transfer by using three protocols: Rsync, Rsync+SSH and Dropbox by comparing 5 cases for each protocol: Case 1, sending individual files of 1MB, 5MB and 10 MB. Case 2, sending 2 files: 1 MB and 5 MB concurrently. Case 3, sending 2 files: 1 MB and 5 MB concurrently where the Client already has 1 MB file. Case 4, sending 3 files 1 MB, 5 MB and 10 MB concurrently. Case 5, sending 3 files 1 MB, 5 MB and 10 MB concurrently in order to test the efficiency of multiple transfer where the Client (com B) already has 1MB and 5 MB files. For all cases, Rsync is the most efficient in terms of time spent in data transfer. Rynsc+SSH requires more time than Rsync but it was much faster than Dropbox.

REFERENCES

[1] D. Bhardwaj and R. Kumar, "A parallel file transfer protocol for clusters and grid system," in *Proc. International Conference on e-Science and Grid Computing*, July 2005, pp. 248-254.

[2] S. Sohail *et al.*, "Parallelized file transfer protocol (P-FTP)," in *Proc. IEEE International Conference on Local Computer Networks*, October 2003, pp. 624-631.

[3] D. J. Barrett and S. E. Richard, *SSH. The Secure Shell: The Definitive Guide*, Cambridge: O' Reilly Media, 2001.

[4] M. G. Jaatun, "Secure file transfer over TCP/IP," in *Proc. IEEE Region 10 International Conference* on *Technology Enabling Tomorrow: Computers, Communications and Automation towards the 21st Century*, November 2008, pp. 494-498.

[5] A. Tridgell and P. Mackeras, "The Rsync algorithm," Technical Report, Australian National University, 1998

[6] U. Irmak, S. Mihaylov, and T. Suel. "Improved single-round protocols for remote file synchronization," in *Proc. Annual Joint Conference of IEEE Computer and Communications Societies*, March 2005, vol. 3, pp. 1665-1676.

[7] I. I. Pamintuan and G. Daniel, "File replication and distribution system for low bandwidth networks," in *Proc. 2002 International Symposium on Parallel Architectures, Algorithms and Networks*, May 2002, pp. 133.

[8] T. Shrivastava, "Rsync(Remote Rsync): 10 Practical Examples of Rsync Command in Linux," *Under Linux Commands*, September 17, 2013.

[9] K. S. Zahed *et al.*, "Reducing storage requirements of snapshot backups based on rsync utility," in *Proc. First International Communication Systems and Networks and Workshops*, Jan. 2009, pp. 1-2.

[10] A. Z. Schwartz "UD dropbox 2.0:collaboration magic," in *Proc. the 35th Annual ACM SIGUCCS Fall Conference*, SIGUCCS'07, 2007, pp. 305-309.

[11] I. Drago *et al*., "Inside dropbox: understanding personal cloud storage services," in *Proc. the 12th ACM SIGCOMM Conference on Internet Measurement*, IMC'12, 2012, pp. 481-494.

[12] H. Yan, U. Irmak, and T. Suel, "Algorithms for low-latency remote file synchronization," in *Proc. INFOCOM 2008, The 27th IEEE Conference on Computer Communications*, 2008, pp. 156-160.

[13] D. Gupta and K. Sagar, "Remote file synchronization single round algorithms," *International Journal of Computer Applications*, vol. 4, issue 1, pp. 32-36, July 2010.

[14] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, issue 1, pp. 1-11, January 2011.

Wiman Banditvilai was born at Chulalongkorn Hospital in Bangkok on November 18, 1986. He got a bachelor of science information technology from Faculty of Liberal Arts and Science, Kasetsart University, Kamphaeng Saen Campus.

He is currently a graduate student at the Department of Information Technology, Faculty of Information Technology at King Mongkut's University of Technology North Bangkok.

Sirapat Boonkrong got B.Sc. and Ph.D. in computer science from University of Bath, United Kingdom. He is currently an assistant professor and associate dean of the Faculty of Information Technology for Academic Affairs and Research at King Mongkut's University of Technology North Bangkok.

He developed an authentication protocol, a pre-hanoff protocol and a hand off protocol for mobile ad hoc networks (MANETs), using C, C++.

His research papers are The Comparison of Impacts to Android Phone Battery between Polling Data and Pushing Data, Outbound Call Route Selection on IP-PBX Using Unstructured Supplementary Service Data, A Step towards a Solution to Information Privacy Problem on Online Social Networks, The Number of Keys Needed for Secure Sharing on Social Networks, The Quest for Security on Online Social Networks, Some Remarks on Andrew Secure RPC.