IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

File Processing in a Distributed Environment

Aayushi Vora[1], Dhruv Porwal[2], Vinaya Sawant[3]

Student, Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India [1, 2]

Assistant Professor, Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India [3]

Abstract: This paper intends to demonstrate the processing of a file in a distributed environment using Remote Method Invocation. The processing will be in terms of splitting, merging and compression, decompression. By using this over a distributed system mainly distributes the functions needed to be performed by the client over the server. It overcomes the limitations of transferring larger files. It splits any file format as per user requirement and merges it without any loss of data. Similarly, it compresses any file format and also decompresses it. Checksum is provided to verify lossless splitting and merging of the file. Also, auto email facility is provided to email the processed file.

Keywords: split; merge; join; checksum; file processing; lossless; compress; decompress; email.

I. INTRODUCTION

First question that comes in our mind is that why should one want to split and recombine files? For instance, think of a file of 50 MB, and try to send it to a friend, post it in a newsgroup or upload it to a Website or FTP server. To reduce memory loading, splitting feature is adapted. It is usually easier to send/receive, upload/download smaller parts than handle the entire file in one go. A file of e.g. 10 GB in size can be split into smaller parts which then can be burned to CD's, DVDs, copied to USB sticks or uploaded to an online backup service.

Once, we have split files that have been be burned to CD's, DVDs, copied to USB sticks or uploaded to an online backup service; you may need to get the original file back. However, this is not possible by opening the split parts because now the splits are in byte format and unreadable too. So to regain the original file we merge the split parts. Merging of split parts means that all the split parts of the file are brought back together to get back the original file with the same size, same quality and same file format too.

This merging is done without any loss of data in the original file. Compression of the file will reduce the file size to a considerable point. Decompression of the file can also be done which will get back the original file. Apart from these functionalities, checksum can be calculated to check if there is loss of data. In case you want to send your processed file over email, auto email facility will enable you to send it. The name of the application developed and implemented by us is PRO-FILE.

II. RELATED WORK

This application has various works which can be related to it. The comparison we made is with GSplit[1], Free File Splitter[2] and HJ-Split[3]. GSplit, Free File Splitter and HJ-Split are stand-alone applications while our application PRO-FILE works in a distributed environment. The Demo File that we splitted is R.S Agarwal (size: 66mb pdf)

Table 1. Comparison of splitting function

Name	File Type Supported	Time to Split	Split Size
GSplit	All	5sec	1mb
Free File Splitter	All	2sec	1mb
HJ-Split	All	2sec	1mb
PRO-FILE	All	3sec	1mb

Table 2 Comparison of various functionalities of applications

Name	Authentication	Checksum	Compression / Decompression	Auto E-mail
GSplit	No	Yes	No	No
Free File Splitter	Yes	No	No	No
HJ-Split	No	Yes	No	No
PRO-FILE	Yes	Yes	Yes	Yes

The screenshots below are HJ-Split User Interface. It provides functionalities like Split, Join, Compare and Checksum.

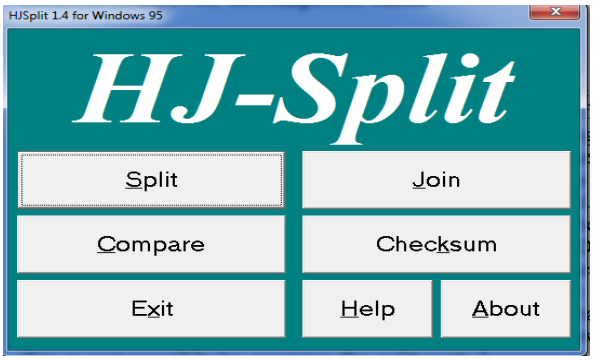

Fig.1. HJ-Split user interface.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

Splitting provided by the application only provides the functionality of splitting by providing the size of each split. Also the accepted size of each split is in Kb. On the other hand, PRO-FILE provides splitting facility where user can either provide size of each split or number of splits. Also, the compression and decompression along with auto email facility are the additional features provided by our software.

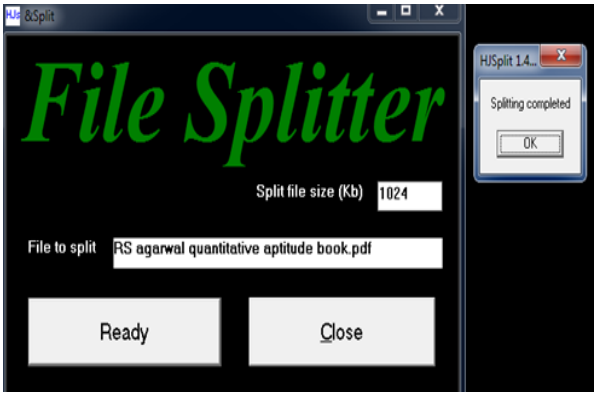

Fig.2. HJ-Split split section user interface.

III. SYSTEM MODEL

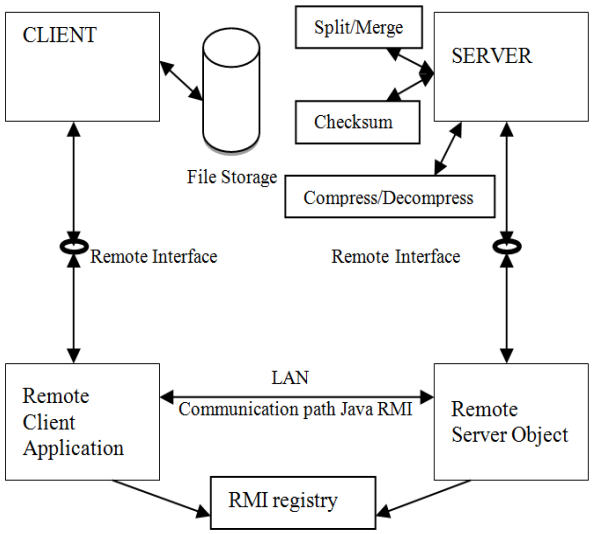

Fig.3. Architectural diagram of the system

The client machine will open the application and browse through the file storage to select a file to be split or merge respectively. Through a remote interface a set of methods which can be invoked by a remote client are declared. The RMI[4] registry runs on the server where the remote object exists and contains a list of references to all remote objects on that server that are accessible to clients through RMI. A client may query the registry to obtain a reference to a remote object. Once the reference is obtained the client can invoke the methods on the remote object through the communication path as if they were local to the client. The server objects implement the methods and return the output to the client machine.

IV. PROBLEM STATEMENT

The problem that the user faces when sending a file which has a large size may be in MB or GB; limitation is being imposed by e-mail providers who only allow you to send files up to some size (25MB). Problem may also occur to fit files into floppy discs or other devices like hard disk or pen drive if you don't have sufficient space to store in it. Also what if a company wants to send data that is large in size along with security; in this case the files which are sent or received need to be confidential. There may be a possibility that the data received by the opposite party is not the same as the one which was sent i.e. data loss occurs. The solution proposed to the problem is a file splitter and merger over a distributed system using Remote method invocation (RMI) provided with checksum to check any kind of data loss and with auto e-mail facility through which we can send the processed file to any desired user with any message attached to it.

V. IMPLEMENTATION

A file splitter can select file of any format and split files into as many parts as desired and also into various sizes as per our requirement. Once the file is split into desired parts or size, the checksum is calculated. The remote client sends the input file to the remote server object which invokes the splitting implementation. Now the split files are sent using a java[5] RMI communication path to the client. Once the split file is received, the split parts are stored in the same directory. The file is encrypted with a password so only the assigned user can access the parts of the file. Now these smaller files can be sent through different means like email, pen drive or hard disk. Now if the other user wants to merge the parts of the file, the file merger is to be used. The file merger will merge the parts into single original file. Before merging the decryption password is to be entered and if right then the files can be accessed.

The merging process happens just like splitting where the parts are sent to the remote server object which invokes the merging implementation and then the checksum of the merged file is calculated. The splitting and merging is lossless so the value of the checksum calculated before splitting and after merging will be the same. The checksum helps you verify if the transfer was lossless. Now the merged file or original file is sent back to the client through java RMI communication path.

The input files and output folder is selected by the user. Based upon the number of parts input by the user or the size for each split; calculations are done automatically. For example, a file selected is of 10MB and the number of parts given by user is 10. Then each part will be of 1MB. .pf is a file that is created after you split any of the file using the application. The .pf file is the first part of the file which is split. The other parts of the file are given extensions like .001, .002, .003 and so on. This file

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

contains all the essential information about split size, number of parts, name of file etc. While merging, we will have to select this .pf file. The data of this file will be read and processed to merge the splits back.

A CRC-enabled device calculates a short, fixed-length sequence, known as the check value, for each block of data to be sent or stored and appends it to the data, forming a code-word. A checksum is a count of the number of bits in a transmission unit that is included with the unit so that the receiver can check to see whether the same number of bits arrived. If the counts match, it's assumed that the complete transmission was received. It is a simple error-detection scheme in which the data has a numerical value based on the number of set bits in the message. The receiving station then applies the same formula to the message and checks to make sure the accompanying numerical value is the same. If not, the receiver can assume that the message has been garbled.

The application PRO-FILE also supports file compression and decompression. Compression[6] is the process of reducing the file size so as to facilitate easy and quick transfer while decompression is the process through which we can get the original file back. This is an additional feature implemented by us as compared to other applications. The file can be compressed using the compress option in which you have to select the file to be compressed. Once you have selected the file, the application names the output file in .pf format. Now you have to select the compression level which ranges from 1 to 9. The value selected determines the size by which the file will be reduced. Once compression is applied to the file we can also check the auto e-mail option to send a direct e-mail to the required address after the file compression is done. The auto e-mail option directly opens the e-mail UI after compression where you add the subject, file, receiver address and any related body message. While on the other end decompression can take place by selecting the compressed .pf file and applying the decompression option on it and selecting the output folder. The decompressed file will thus be stored in the same folder as specified.

The auto e-mail facility in our program uses classes like Auto_Email_Client, MailProjectClass. Gmail provider is used using SMTP protocol port number 587.Multipart message is sent that is including body, receiver e-mail address and subject. TLS is used for security purpose. It will display if the e-mail is sent or not. Auto_Email_Client class attaches the compressed file automatically. The path of the file to be attached is taken from the Output file text box. This class is also used for the UI purpose. The MailProjectClass basically creates the session between client and server and uses the port number. It authorizes the username and password of the sender. Transport send(message) method is used to send the e-mail. This will take the message as combination of receiver address, subject and body.

Note: This *.pf file should be in the same folder as the other split parts in order to merge.

VI. SCREENSHOTS

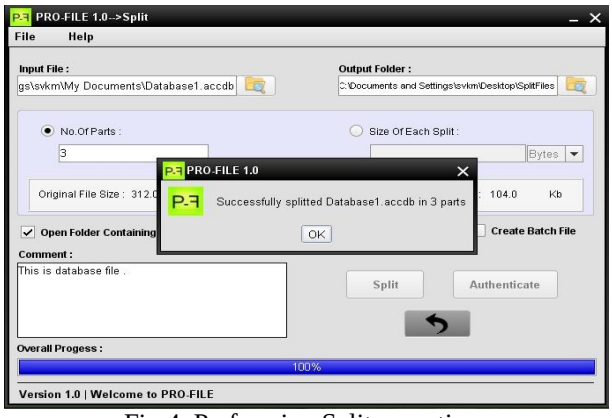

Fig.4. Performing Split operation.

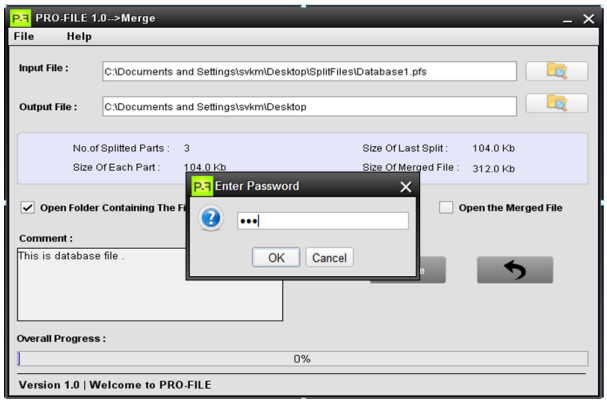

Fig.5. Performing Merge operation.

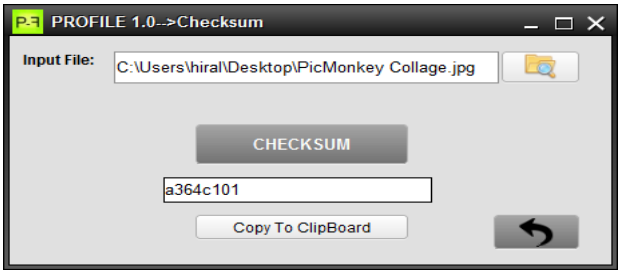

Fig.6. Performing Checksum operation.

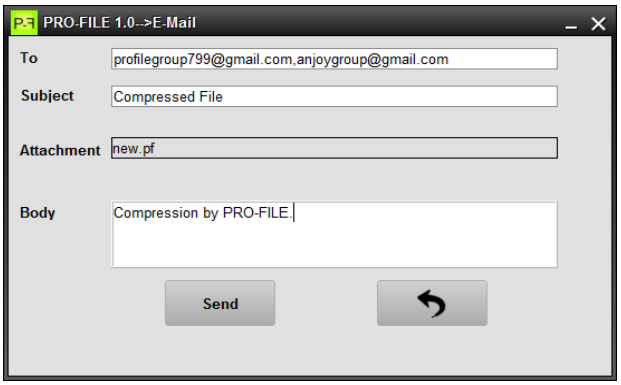

Fig.7. Performing Auto E-mail operation.

DOI 10.17148/IJARCCE.2016.511104

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

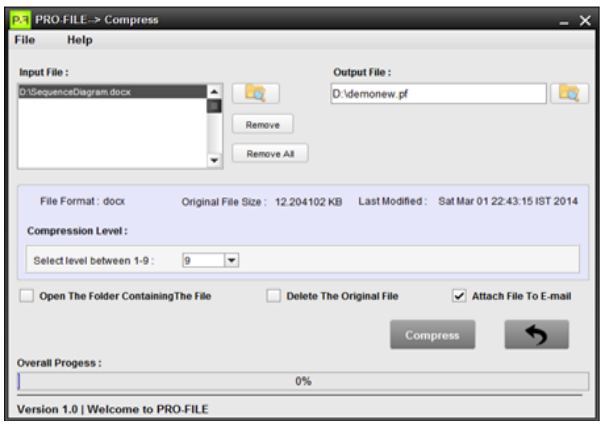

Fig.8. Performing Compression operation

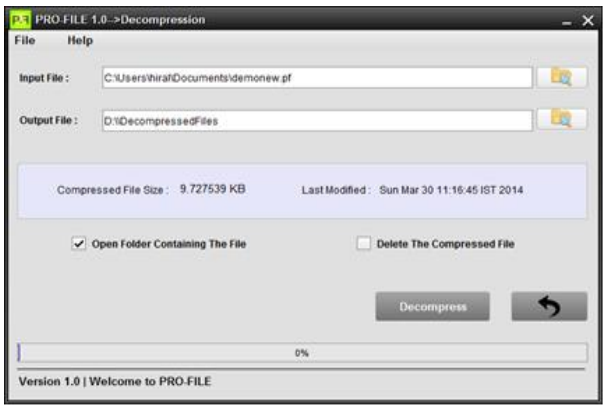

Fig.9. Performing Decompression operation

VII. ALGORITHMIC REPRESENTATION

1. SPLIT

Step 1.Start
Step 2.Read file name "filename".
Step 3.Read file size.
Step 4.Read number of parts "number_parts"/ Read size of each split "split_size".
Step 5.If filename_length < split_size then Error

Else
Create New Directory "Splitted_Files"
Call_Split(filename,destination_filename,comments, number_parts, split_size,last_split_size,delete,batch)
Step5.1 Read filename
Step5.2 Create filename.pf
Step5.3 Encrypt data in filename.pf file
Step5.4 Divide byte data of the file in new Split files created
Step6.Open Splitted_Files with all split files and filename.pf file
Step 7.Close

2. MERGE

Step 1.Start
Step 2.Read the filename.pf file
Step 3.Check if all parts are available in the directory
Step 4.Call Merge(filename,destination)
Step 4.1 Read filename.pf
Step 4.2 Decrypt the contents pf filename.pf
Step 4.3 Merge the files by processing filename.pf
Step 5.Close

3. COMPRESSION

Step 1. Import util.zip package
Step 2. Select the file "filename"
Step 3. Call the zip (filename, compression_path)
Step 3.1 Create object of ZipOutputStream and get the zip file as output
Step 3.2 Use the Stack array and check if the file already existing in the determined folder.
Step 4. Save the compressed file.

4. DECOMPRESSION

Step 1. Import util.zip package
Step 2. Select the file "filename"
Step 3. Call the unzip (filename, decompression_path)
Step 3.1 Create object of ZipInputStream and get the original file as output
Step 4. Save the decompressed file.

VIII. ACKNOWLEDGEMENTS

We are thankful to **Dr. Abhijit Joshi,** Vice principal (Acad.) of Dwarkadas J. Sanghvi College of Engineering and Head of Department of Information Technology. His guidance and insights for the paper was useful. We are also grateful to our peers, staff members and experts of this department for their support in this research and writing this manuscript.

REFERENCES

[1] GSplit software : http://www.gdgsoft.com/gsplit/index , referred on 20th October,2016.
[2] Free File Splitter : http://www.filesplitter.org/ , referred on 25th October,2016.
[3] HJ-Split : http://www.hjsplit.org/docs/online_manual/ , referred on 1st November 2016.
[4] George Coulouris, Jean Dollimore, Tim Kindberg's Distributed Systems Concepts and Design,Third Edition by Pearson Education.
[5] Herbert Schildt, The Complete Reference Java, Seventh edition.
[6] Prabhat K. Andleigh and Kiran Thakrar, Multimedia Systems Design, Eastern Economy Edition.