MULTITHREADED DOWNLOAD MANAGER

A PROJECT REPORT

SUBMITTED BY:

Ayush Sharma (19BCE0408)

Anant Tiwari (19BCE2178)

Keshav Dalmia (19BCE2148)

Harshit Vijay (19BCE2321)

COURSE CODE: CSE 4001

COURSE TITLE: PARALLEL AND DISTRIBUTED COMPUTING

Under the guidance of

Dr. R Balamurugan

Associate Professor, SCOPE,

VIT, Vellore.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX Page no.

ABSTRACT

Downloading a file is a frequently worked domain where we use the default download managers of a given browser to download a file. While most download managers download the file from the same source, some downloading managers can likewise increase the download speeds by downloading from different sources simultaneously. Despite the fact that browsers may have download administrators in built as an element, they are separated by the way that they don't organize exact, finish and unbroken downloads of data. While some download administrators are completely fledged projects devoted to downloading any data more than at least one convention (e.g. http), many are incorporated into installers or refresh directors and used to download parts of a particular program (or set of projects).

eg. contains Google and Adobe's update.

1. INTRODUCTION

Download managers were among the top (includes torrent customers as they are actually download managers also) applications showing a flag promotion in the UI. Many download managers accompany the highlights like video and sound retrieving from well-known websites like YouTube etc., They likewise support site snatching. queue handling is another main element of download administrator. They additionally can pause and resume downloads, and force speed confinements too. This highlight come exceptionally helpful in locales where power failures is an issue. Furthermore, a large portion of the business download chiefs can download following client arranged timetables and download in like manner. A couple of download administrators claim to build the download speed by a factor of ordinarily. Download managers additionally have tight integration with engines. For the most part they do this by introducing an expansion to the client's engine(browser). Download speeding up, otherwise called multipart download, is a term for the strategy utilized by programming, for example, download administrators to download a solitary record by part.

2. LITERATURE SURVEY

2.1. PAPERS

PAPER – I:

TITLE: Thread Assignment of Multithreaded Network Applications in Multicore/ Multithreaded Processors

AUTHOR: Petar Radojkovic, Vladimir Cakarevic, Javier Verdu, Alex Pajuelo, Francisco J. Cazorla

JOURNAL NAME &DATA: IEEE Transactions on Parallel and Distributed Systems 12, Dec.2013

KEY CONCEPTS: Thread assignment, Lightweight Kernel (LWK), Blackbox scheduler

ADVANTAGES: Systematic method for thread assignment of multithreaded network applications running on multicore/multithreaded processors.

DISADVANTAGES: Proposed method is evaluated with an industrial case study for a set of multithreaded network applications

FUTURE IMPROVEMENT: Finding optimal thread assignment on multithreaded processors comprising many cores is an NPcomplete problem

PAPER – II:

TITLE: Effect of thread level parallelism on the performance of optimum architecture for embedded applications

AUTHOR: Mehdi Alipour , Hojjat Taghdisi

JOURNAL NAME &DATA: International al Journal of Embedded Systems and Applications (IJESA) Vol.2, No.1, March 2012

KEY CONCEPTS: Thread Level Parallelism, Optimum single-thread architecture

ADVANTAGES: Running two threads on a single thread processor with limited area and power budget, in average, Leads to increased performance for some representative embedded applications.

DISADVANTAGES: Negative impact of parallel downloading on Processor framework.

PAPER – III:

TITLE: Proceedings in the Third IEEE Workshop on Internet Applications

AUTHOR: Gkantsidis, M. Ammar and E. Zegura.

JOURNAL NAME &DATA: IEEE WIAPP 2003

KEY CONCEPTS: Functionality of parallel downloading When grasped inside the server. Survey on different type of parallel downloading plans &customers point of view on their execution.

ADVANTAGES: Clients easily identify the execution change between multithread parallel downloading and single processor-based downloading.
DISADVANTAGES: Parallel downloading brings extra overhead due to extra use of system resources.

PAPER – IV:

TITLE: Operating systems with Parallel threading
AUTHOR: Zhou Xu, Lu Xianliang, Hou Mengshu and Zhan Chaun
JOURNAL NAME &DATA: ACM SIGOPS, Operating, Systems review, Homepage 2005
KEY CONCEPTS: Advance parallel downloading is a cutting-edge parallel download plan in P2P condition. It also tell about initial investment of threads for each downloading server.
ADVANTAGES: After speedier servers finish their own specific work, parallel downloading reallocates some part of their work to deficient server who works slow.
DISADVANTAGES: Due to allocation of threads to slower server, it produces immense pressure on processors.

PAPER – V:

TITLE: Parallel Downloading Using Variable Length Blocks for Proxy Servers
AUTHOR: Atsushi Kawano, Junichi Funasaka, Kenji Ishida
JOURNAL NAME &DATA: 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07) - 30 July 2007
KEY CONCEPTS: Proposed a proxy server which gets the latest records at rapid speed by utilizing the parallel downloading innovation
ADVANTAGES: Rapid speed. Packets are reordered and sent to the customer
DISADVANTAGES: However, this proxy server has the issue that the cushion may turn out to be too huge.
FUTURE IMPROVEMENT: The piece measure is resolved powerfully as indicated by the system condition with the goal that the involved cushion space contracts.

PAPER – VI:

TITLE: Multithreaded Pipeline Synthesis for Data Parallel Kernels

AUTHOR: Mingxing Tan1 , Bin Liu2 , Steve Dai1 , Zhiru Zhang

JOURNAL NAME &DATA: 2014 IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2- 6 Nov. 2014

KEY CONCEPTS: Multithreaded pipelining approach that enables context switching to allow out-of-order thread execution for data parallel kernels.

ADVANTAGES: Proposed techniques can significantly improve the effective pipeline throughput over conventional approaches

DISADVANTAGES: The hardware overhead associated with context management is significantly more.

PAPER – VII:

TITLE: Topology-aware server selection method for dynamic parallel downloading

AUTHOR: Y. Higashi , S. Ata

JOURNAL NAME &DATA: Second IEEE Consumer Communications and Networking Conference 2005

KEY CONCEPTS: Downloading by taking the physical network topology into consideration

ADVANTAGES: Approach tries to minimize the total number of shared links between a client and downloading servers.

DISADVANTAGES: If some connections share the same bottleneck link, the throughput is not increased, and the additional connection simply wastes the server resources.

FUTURE IMPROVEMENT: Provide a way to virtually handle cases when a bottleneck arises due to physical topology and decide whether parallel is a better approach

PAPER – VIII:

TITLE: Characterizing the Performance and Energy Efficiency of Simultaneous Multi-threading in Multicore Environments

AUTHOR: Balaji Subramaniam and Wuchun Feng.

JOURNAL NAME &DATA: 41st International Conference on Parallel Processing Workshops 2012

KEY CONCEPTS: Concurrency, Multicore processing, Multithreading, simultaneous multithreading

ADVANTAGES: Focus on concurrent throttling. Improving the processor utilization by simultaneous multithreading

DISADVANTAGES: Benefit of concurrency throttling is highly dependent on the workload

PAPER – IX:

TITLE: Characterizing the Performance and Energy Efficiency of Simultaneous Multi-threading in Multicore Environments

AUTHOR: Balaji Subramaniam and Wuchun Feng.

JOURNAL NAME &DATA: 41st International Conference on Parallel Processing Workshops 2012

KEY CONCEPTS: Concurrency, Multicore processing, Multithreading, simultaneous multithreading

ADVANTAGES: Focus on concurrent throttling. Improving the processor utilization by simultaneous multithreading

DISADVANTAGES: Benefit of concurrency throttling is highly dependent on the workload

PAPER – X:

TITLE: Efficient Development Methodology for Multithreaded Network Application

AUTHOR: Nguyen Duc Chinh, Ettikan Kandasamy, Lam Yoke Khei

JOURNAL NAME &DATA: The 5th Student Conference on Research and Development December 2007

KEY CONCEPTS: Pitfalls in multithreaded applications

ADVANTAGES: Discovery of common pitfalls in multithreaded network applications. Comparison of performance of a single threaded network application with multithreaded network applications.

DISADVANTAGES: Difficulty in detection of thread related errors like race errors, deadlocks

2.2. PROBLEM DEFINITION

- Improvement of performance by the multithreaded manager
- Improvement of processor utilization by simultaneous threading
- Enabling the ease of using download manager, even when the power goes off, the download is resumed from there
- If a balance is found between hardware overhead and improvement in parallelism in kernel then it could be scalable to a huge level

3. OVERVIEW OF THE WORK

3.1. OBJECTIVES OF PROJECT

- Improve the download speed by creating download manager
- Provide options to pause ,resume, and cancel download
- To provide user friendly download manager which pauses the download whenever the power or the internet goes off
- Use multithreading to create GUI's frames which has the information regarding Download like progress bar , file name, file location, etc.

3.2. SOFTWARE REQUIREMENTS

For the proposed system, we shall be using the following software and platforms for development purpose:

- Platform – Net Beans IDE
- An open source development environment
- Programming language - JAVA
- We need the Java Development Kit with the Java compiler

There are several sources for a JRE/JDK. Here are some of the more common/popular ones:

- IBM JDK
- Open JDK
- Oracle JDK

NETBEANS:

- NetBeans is an open-source integrated development environment (IDE) for developing with java, php, c++, and other programming languages.
- NetBeans is also referred to as a platform of modular components used for developing java desktop applications.
- NetBeans aims to help programmers develop software.
- NetBeans contains both a compiler and an interpreter, and switches between them according to need.
- NetBeans supports many languages and comes with a variety of plugins

4. SYSTEM DESIGN

4.1. ALGORITHMS, BLOCK DIAGRAMS

- A thread is created for IDM which creates a thread for Download manager.
- Another thread is created for progress panel which monitors the progress of the download.
- When download is complete, a new action is executed in the main thread which stops the 2 running threads and executes the downloadcomplete thread.
- Downloadcomplete class is executed with the help of a thread which displays the final download complete screen which displays the downloaded file.
- We have followed Object-Oriented approach to build this project in JAVA. We have divided our tasks into sub-modules, each module performing some specific functionality. These modules work together and download a file
- By splitting the download process into multiple blocks and assigning a thread to each block which opens up a connection with the server and finally when all threads are finished, all threads are joined to get the final downloaded file.

4.2. USE OF MULTICORES IN JAVA VIRTUAL MACHINE

The standard JVM runs parallel Java threads in parallel, runs multiple threads for Garbage Collection and also the JIT compiler, and runs a number of other java-level but not user threads for a bunch of housekeeping tasks.

The JVM is well tested and proven on hundreds of cores with thousands of runnable threads. The JVM can be “pinched” down to a specific number of cores by any OS which can limit CPUs (e.g. Linux cogroups), as long as the OS also multi-tasks the runnable threads on the available cores.

The Azul/Zing JVM runs well on ~1000 cores and ~100K runnable threads (so 10x more cores and 100x more runnable threads).

The Java Runnable interface is designed for handling parallel and concurrent programs. Any class that implements this interface must implement this function:

```
@Override
public void run()
{
//implementation here
}
```

5. IMPLEMENTATION

5.1. DESCRIPTION OF MODULES/PROGRAMS

i) Downloader.java

a) Creates map(a type of list) of threads
b) declares current state of the download as one of the following: PAUSED,DOWNLOADING or CANCELED
c) Taking the URL and type conversion to string, fetching the size of the file.
d) Calculating the progress (%) using the formula (Downloaded / File Size) * 100
e) Set the state of the download
f) Start or resume the download
g) Increase the downloaded size
h) Set the state has changed and notify the observers
i) Create thread to download a part of file
j) See whether the thread is finished or not, wait for the thread to finish

ii) IDM.java

a) Get and set the maximum number of connections possible per download(This is essentially the service provided by the downloading website)
b) Get the downloader object in a list(the object from previous class)
c) Get the download list
d) Verify if the URL is valid

iii)Progresspanel.java

a) Create an instance of DownloadTableModel
b) Set up the table
c) Allow only one row at a time to be selected
d) Set up progress bar as progress renderer for progress column
e) Set table's row height large enough to fit progress bar
f) Create button, text fields, scroll pane and table

g) Set the text for title bar, for jbnAdd button set the text to "Add Download", for jbnPause button set the text to "Pause", for jbnRemove button set the text to "Remove", for jbnRemove button set the text to "Remove", for jbnResume button set the text to "Resume", for jbnExit button set the text to "Exit"

h) Create a layout and add these components to the layout specifying the dimensions of every component

i) Assign the function to the button

j) If buttons are pressed then change the state of the button

iv) DownloadCompleteFrame.java

a) Initializes various components of the progress check function

b) Reflects the progress of download by showing percentage of file downloaded

c) Invokes startThread(IDM) to create and stop new thread for download

d) JSwing is used to develop the GUI and various functions like JPanel,JFrame,JButton are used for development

5.2. SOURCE CODE

a) IDM.java

```
package IDMPackage;
import java.io.*; //Input Output data streams
import java.net.URL;
import java.net.HttpURLConnection;
import java.nio.charset.MalformedInputException;
import java.text.DecimalFormat; //Use to format numbers using a formatting pattern specified by us
import java.util.Date; //Used to display date and time
import java.util.logging.Level; //recording application activity - warning,exception information
import java.util.logging.Logger; //recording application activity - warning,exception information
public class IDM implements Runnable {//instance intended to be executed by thread
private String urlName;
private String OutputFileName;
 int threadNumber;
```

```
private URL url;
private HttpURLConnection con;
private BufferedInputStream input;
public BufferedOutputStream output;
private byte[] buffer;
int downloaded;
int totalLength;
int startPos = 0;
int endPos = 0;
int pos = 0;
public boolean threadState;
public static int progress = 0;
ProgressPanel panel;
int b;
Date date;
int bread = 0;
int bytesPerSec = 0;
public IDM(String url, ProgressPanel panel, String directory)
{ date = new Date();
//b = date.getSeconds();
threadState = true;
this.urlName = url;
this.OutputFileName = directory + "\\" + url.substring(url.lastIndexOf('/') + 1);
try {
System.out.println("full directory with file name: " + OutputFileName);
this.url = new URL(urlName);
this.con = (HttpURLConnection) this.url.openConnection();
input = new BufferedInputStream(con.getInputStream());
totalLength = con.getContentLength();
downloaded = 0;
output = new BufferedOutputStream(
new FileOutputStream(OutputFileName));
buffer = new byte[2 * 1024];
this.panel = panel;
```

```
 } catch (MalformedInputException malformedInputException)
 { malformedInputException.printStackTrace();
 } catch (IOException ioException)
 { ioException.printStackTrace();
 }
 }
 public void setState(boolean state)
 { threadState = state;
 }
 public int getDownloaded()
 { return downloaded;
 }
 @Override
 public void run()
 { int bytesRead;
 try {
 while ((bytesRead = this.input.read(buffer)) != -1)
{ endPos = bytesRead;
new IDM.ReadThreadClass(buffer, 0, bytesRead);
System.out.println("Pos: " + startPos + ":" + endPos);
startPos = endPos;
this.output.flush();
 }
 this.output.flush();
 this.input.close();
 this.output.close();
 } catch (MalformedInputException malformedInputException)
 { malformedInputException.printStackTrace();
 } catch (IOException ioException)
 { ioException.printStackTrace();
 }
 }
 ProgressPanel oo = new ProgressPanel();
 class ReadThreadClass implements Runnable {
```

```
private int startPs, endPs;
byte[] buffewr;
public ReadThreadClass(byte[] buffer, int startPos, int endPos)
{ startPs = startPos;
endPs = endPos;
buffewr = buffer;
run();
}
public ReadThreadClass(int bytesRead)
{ endPs = bytesRead;
}
boolean a = false;
@Override
public void run()
{ try {
output.write(buffer, 0, endPs);
downloaded += endPs;
double downloadedd = ((double) downloaded / totalLength) * 100;
progress = (int) downloadedd;
panel.setProgressValue(progress);
System.out.println("progress: " + progress);
double percentage = Double.parseDouble(new DecimalFormat(".##").format(downloadedd));
System.out.println("downloaded: " + downloaded + " : " + (percentage) + "%");
panel.setTextToArea(totalLength, percentage);
} catch (IOException ex)
{ Logger.getLogger(IDM.class.getName()).log(Level.SEVERE, null,
ex);
}
}
}
}
```

b) Progresspanel.java

```
package IDMPackage;
public class ProgressPanel extends javax.swing.JFrame
   { Thread thread;
   String outputDirectory;
   private String outputFile;
   public ProgressPanel() {
      initComponents();
   this.setVisible(true);
   }
      public void setProgressValue(int value)
   { progressBar.setValue(value);
   if (progressBar.getPercentComplete() == 1.0)
   { System.out.println("FINISHED");
   DownloadCompleteFrame dcf = new DownloadCompleteFrame(outputDirectory,
urlTextField.getText(), outputFile);
 }
 }
   public void setUrlTextField(String url)
   { urlTextField.setText(url);
   }
   public void startThread(IDM idm)
   { thread = new Thread(idm);
   thread.start();
 }
   public void stopThread()
   { this.thread.stop();
 }
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {

      jLabel1 = new javax.swing.JLabel();
```

```
progressBar = new javax.swing.JProgressBar();
pauseButton = new javax.swing.JButton();
resumeButton = new javax.swing.JButton();
cancelButton = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
urlTextField = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jSeparator2 = new javax.swing.JSeparator();
jScrollPane1 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();
jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel1.setText("Progress");

progressBar.setForeground(new java.awt.Color(102, 255, 102));
progressBar.setStringPainted(true);

pauseButton.setFont(new java.awt.Font("Calibri", 3, 18)); // NOI18N
pauseButton.setText("Pause");
pauseButton.addActionListener(new java.awt.event.ActionListener()
    { public void actionPerformed(java.awt.event.ActionEvent evt) {
        pauseButtonActionPerformed(evt);
    }
});

resumeButton.setFont(new java.awt.Font("Calibri", 3, 18)); // NOI18N
resumeButton.setText("Resume");
resumeButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
        { resumeButtonActionPerformed(evt);
    }
```

```
});

cancelButton.setFont(new java.awt.Font("Calibri", 3, 18)); // NOI18N
cancelButton.setText("Cancel");
cancelButton.addActionListener(new java.awt.event.ActionListener()
    { public void actionPerformed(java.awt.event.ActionEvent evt) {
        cancelButtonActionPerformed(evt);
    }
});

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel3.setText("URL");

urlTextField.setEditable(false);
urlTextField.setMaximumSize(new java.awt.Dimension(100, 20));
urlTextField.addActionListener(new java.awt.event.ActionListener()
    { public void actionPerformed(java.awt.event.ActionEvent evt) {
        urlTextFieldActionPerformed(evt);
    }
});

jScrollPane1.setViewportView(jTextArea1);

jTextArea1.setColumns(20);
jTextArea1.setRows(5);
jScrollPane1.setViewportView(jTextArea1);

jLabel2.setForeground(new java.awt.Color(255, 0, 0));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
    ADING)
    .addGroup(layout.createSequentialGroup()
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(80, 80, 80)
                    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(urlTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 549,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(81, 81, 81)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE, 549,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(100, 100, 100)
                    .addComponent(pauseButton, javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(135, 135, 135)
                    .addComponent(resumeButton)
                    .addGap(129, 129, 129)
                    .addComponent(cancelButton, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 406,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 364,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(77, 77, 77)
```

```
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 614,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(169, 169, 169)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 585,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
      DING)
      .addGroup(layout.createSequentialGroup()
        .addGap(70, 70, 70)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(3, 3, 3)
            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addComponent(urlTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(39, 39, 39)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(pauseButton)
          .addComponent(resumeButton)
          .addComponent(cancelButton))
        .addGap(6, 6, 6)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 319,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(38, 38, 38)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(83, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void pauseButtonActionPerformed(java.awt.event.ActionEvent evt) {

        this.thread.suspend();
    }

    private void resumeButtonActionPerformed(java.awt.event.ActionEvent evt) {

        this.thread.resume();
    }

    private void cancelButtonActionPerformed(java.awt.event.ActionEvent evt)
        { stopThread();
        this.jLabel2.setText("Download has been cancelled");
    }
    private void urlTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
    }

    // Variables declaration - do not modify
    private javax.swing.JButton cancelButton;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JButton pauseButton;
    private javax.swing.JProgressBar progressBar;
    private javax.swing.JButton resumeButton;
    private javax.swing.JTextField urlTextField;
    // End of variables declaration

    void setTextToArea(int b, double c) {
    jTextArea1.append("Total Size :" + (float) b / 1024 + " kbs " + " : percentage:" + c + "%" + "\n");
 }
    void setOutputDirectory(String string, String file)
    { outputDirectory = string;
    outputFile = file;
 }
}
```

c) downloader.java

```
package IDMPackage;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import javax.swing.*;
public class downloader extends javax.swing.JFrame implements
    ActionListener{ static void downloadComplete() {
 }
 IDM idm;
 Thread thread;
 ProgressPanel progressPanel;
```

```
JLabel label;
public static JFrame downloaderFrame;
  public downloader() {
     initComponents();
  }

  @SuppressWarnings("unchecked")
  // <editor-fold defaultstate="collapsed" desc="Generated Code">
  private void initComponents() {
     jLabel1 = new javax.swing.JLabel();
     urlTextField = new javax.swing.JTextField();
     downloadButton = new javax.swing.JButton();
     jButton1 = new javax.swing.JButton();
     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
     jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
     jLabel1.setText("URL : ");
     urlTextField.addActionListener(new java.awt.event.ActionListener()
        { public void actionPerformed(java.awt.event.ActionEvent evt) {
           urlTextFieldActionPerformed(evt);
        }
     });

     downloadButton.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
     downloadButton.setText("Save To");
     downloadButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt)
           { downloadButtonActionPerformed(evt);
        }
     });

     jButton1.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
     jButton1.setText("Close");
     jButton1.addActionListener(new java.awt.event.ActionListener()
        { public void actionPerformed(java.awt.event.ActionEvent evt)
        {
```

```
            jButton1ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
        ADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(110, 110, 110)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(urlTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 532,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(222, 222, 222)
            .addComponent(downloadButton, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(47, 47, 47)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );
    layout.setVerticalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
        DING)
        .addGroup(layout.createSequentialGroup()
            .addGap(76, 76, 76)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addGroup(layout.createSequentialGroup()
                    .addGap(2, 2, 2)
                    .addComponent(urlTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(29, 29, 29)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
                .addComponent(downloadButton)
                .addComponent(jButton1))
            .addContainerGap(14, Short.MAX_VALUE))
        );

        urlTextField.getAccessibleContext().setAccessibleName("urlTextField");

        pack();
    }// </editor-fold>
    private void urlTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
    }
    private void downloadButtonActionPerformed(java.awt.event.ActionEvent evt)
        { JFileChooser fileChooser = new JFileChooser();
        fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
        int result = fileChooser.showSaveDialog(this);
        File file = fileChooser.getSelectedFile();
        System.out.println("directory: " + file.getPath());
        progressPanel = new ProgressPanel();
         try {
        idm = new IDM(urlTextField.getText(), progressPanel, file.getPath());
        System.out.println("Starting Executers");
        progressPanel.startThread(idm);
        progressPanel.setUrlTextField(urlTextField.getText());
        progressPanel.setOutputDirectory(file.getPath(),
urlTextField.getText().substring(urlTextField.getText().lastIndexOf('/') +1));
        urlTextField.setText(urlTextField.getText());
        this.dispose();
 } catch (Exception e) {
 }
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
        { this.dispose();
    }
    public static void main(String args[]) {
```

```
        java.awt.EventQueue.invokeLater(new Runnable()
            { public void run() {
                new downloader().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton downloadButton;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JTextField urlTextField;
    // End of variables declaration
    @Override
    public void actionPerformed(ActionEvent evt) {
 }
}
```

d) DownloaderCompleteFrame.java

```
package IDMPackage;
import java.awt.Desktop;
import java.io.File;
import java.io.IOException;
public class DownloadCompleteFrame extends javax.swing.JFrame {
     public DownloadCompleteFrame(String outputDirectory,String url,String outputFile)
    { initComponents();
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    this.jLabel1.setText(url);
    this.jLabel3.setText(outputFile);
    this.jLabel5.setText(outputDirectory);
    this.setVisible(true);
 }
```

```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Download Complete");
    jLabel1.setName("UrlLabel"); // NOI18N
    jLabel2.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
    jLabel2.setText("URL");

    jLabel3.setName("fileNameLabel"); // NOI18N
    jLabel4.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
    jLabel4.setText("File Name");
    jLabel5.setName("fileDirectory"); // NOI18N
    jLabel6.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
    jLabel6.setText("File Directory");
    jButton1.setFont(new java.awt.Font("Calibri Light", 1, 18)); // NOI18N
    jButton1.setText("Open File");
    jButton1.addActionListener(new java.awt.event.ActionListener()
        { public void actionPerformed(java.awt.event.ActionEvent evt)
        {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setFont(new java.awt.Font("Calibri Light", 1, 18)); // NOI18N
```

```
        jButton2.setText("Close");
        jButton2.addActionListener(new java.awt.event.ActionListener()
            { public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
            ADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(92, 92, 92)
                        .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 611,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(175, 175, 175)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(114, 114, 114)
                                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 339,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jLabel6)
                                .addGap(93, 93, 93)
                                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 272,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jLabel4)
```

```
                    .addGap(110, 110, 110)
                    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 272,
javax.swing.GroupLayout.PREFERRED_SIZE)))))
            .addGap(0, 77, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(204, 204, 204)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(143, 143, 143)
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
        DING)
        .addGroup(layout.createSequentialGroup()
            .addGap(103, 103, 103)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(48, 48, 48)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(62, 62, 62)
                    .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
```

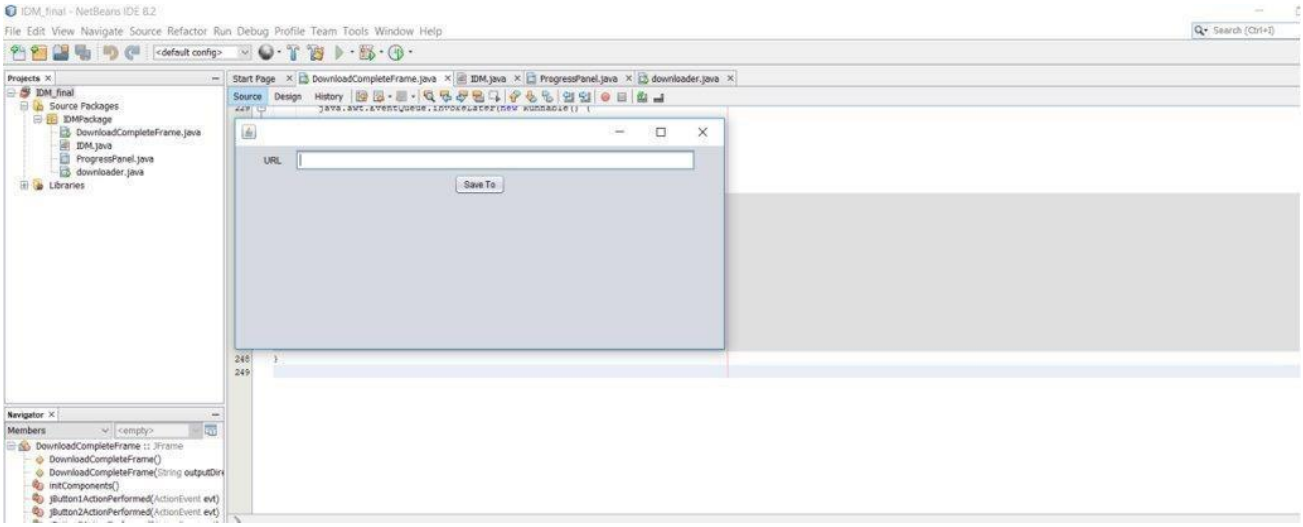
```
                .addGap(52, 52, 52)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(42, 42, 42)
            .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(28, 28, 28)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jButton2)
              .addComponent(jButton1))
            .addContainerGap(64, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
        { Desktop desktop = Desktop.getDesktop();
try {
desktop.open(new File(this.jLabel5.getText()));
} catch (IOException e){ }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
        { this.dispose();
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JSeparator jSeparator1;
```

```
    // End of variables declaration
}
```

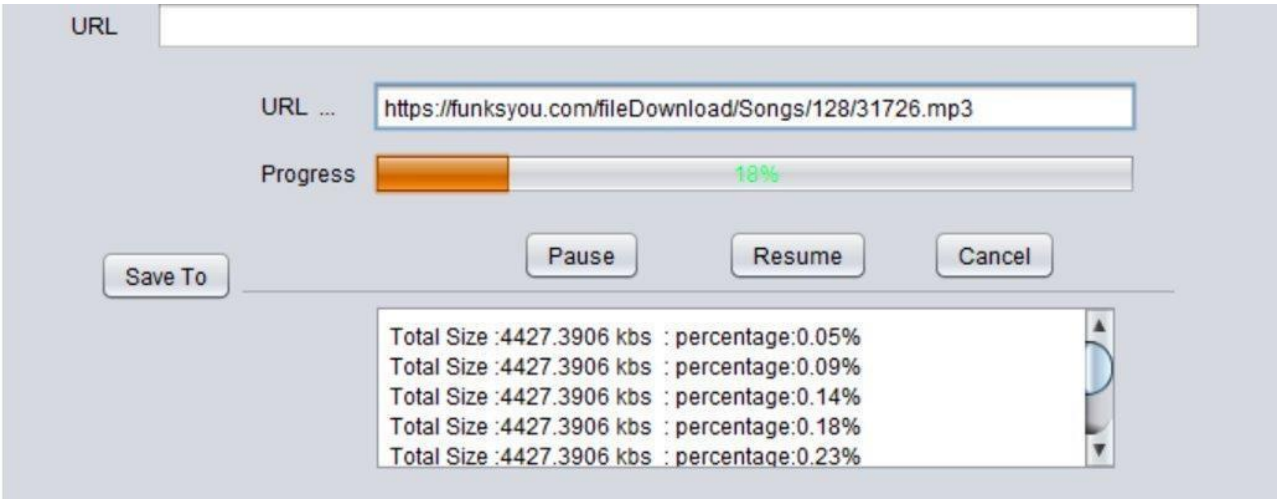
6. OUTPUT AND PERFOMANCE ANALYSIS

6.1. EXECUTION SNAPSHOTS

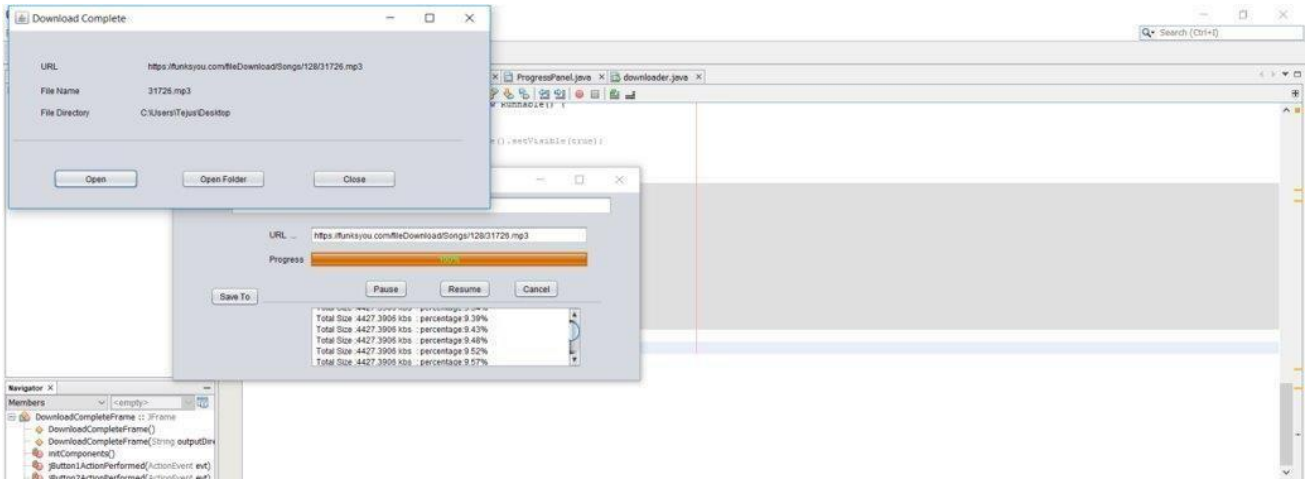


6.2. OUTPUT – IN TERMS OF PERFORMANCE METRICS

UPPON ADDING URL AND SETTING A LOCATION TO SAVE:

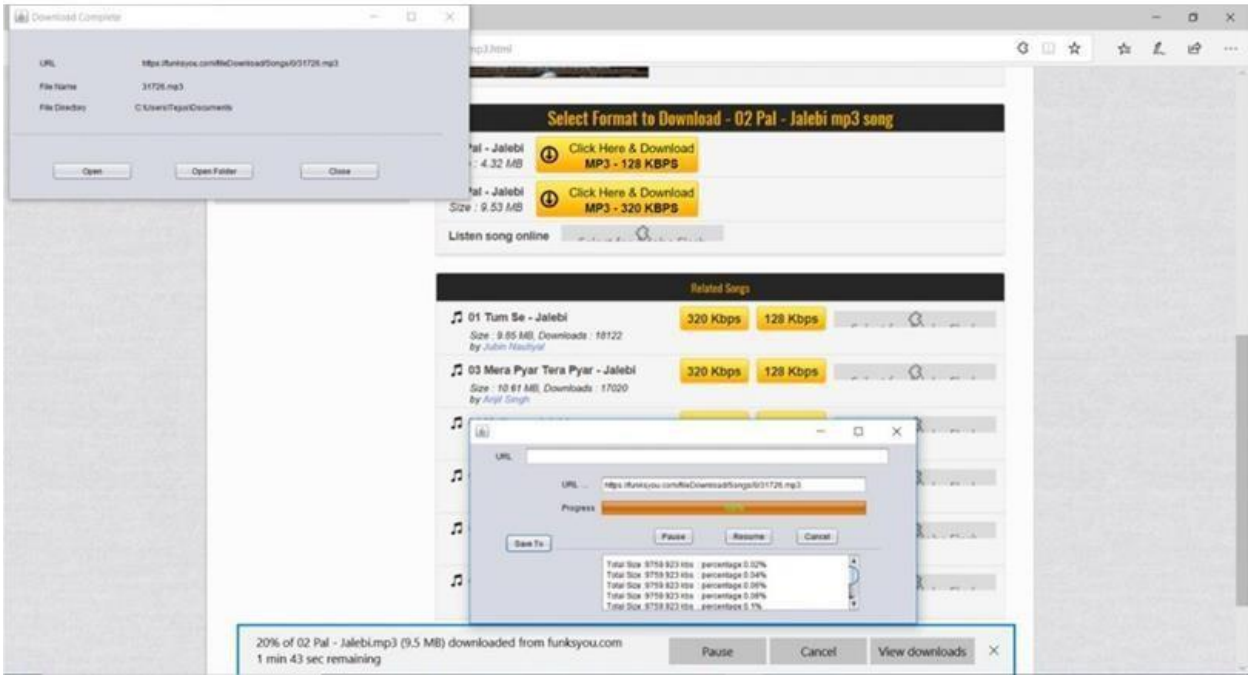


WHEN THE DOWNLOAD IS COMPLETE:

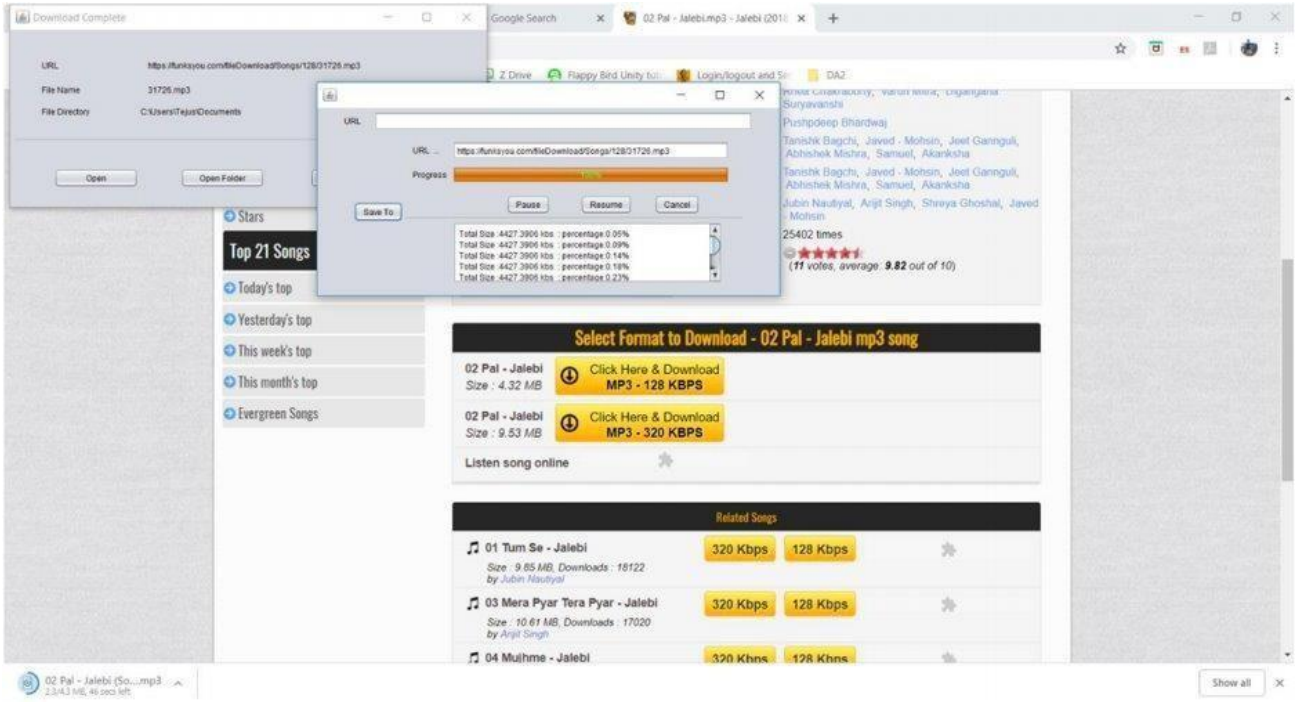


6.3. PERFORMANCE COMPARISON WITH EXISTING WORKS

OUTPUT(COMPARISON WITH MICROSOFT EDGE):



OUTPUT(COMPARISON WITH GOOGLE CHROME):



The above Chart shows the speed up achieved on running the download using our multithreaded download manager in comparison to using Google Chrome and Microsoft Edge browser download managers. It can be seen that a speedup can be achieved using our model. Also, in case of disconnectivity of internet, our downloader resumes the download from the point of dysconnectivity whereas other browsers cancel the download

7. CONCLUSION AND FUTURE DIRECTIONS

We have presented a parallel downloading method using multi-threading in JAVA. The performance is better than serial downloading as the bandwidth is utilized more efficiently. We have compared the time taken for different download managers and came to a conclusion that multi-threaded downloading is faster. In future, we would like to add more features to the application in the future like download scheduling, bandwidth allocation etc.

8. REFERENCES

[1] C. Gkantsidis, M. Ammar and E. Zegura. "Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003".

[2] S.G.M. Koo, C. Rosenberg and Dongyan Xu. "The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings."

[3] Zhou Xu, Lu Xianliang, Hou Menasha and Zhan Chaun. " ACM SIGOPS Operating Systems Review Homepage archive Volume 39 Issue 1, January 2005 Pages 63-69 "

[4] Jiantao Song, Chaofeng Sha and Hong Zhu. "Distributed Computing Systems, 2004. Proceedings."

[5] Allen Miu and Eugene Shih. "Laboratory of Computer Science Massachusetts Institute of Technology Cambridge, MA, USA"

[6] J. Funasaka, N. Nakawaki and K. Ishida. "Distributed Computing Systems Workshops, 2003. Proceedings."

[7] Atsushi Kawano, Junichi Funasaka and Kenji Ishida. "Distributed Computing Systems Workshops, 2007. ICDCSW '07."

[8] Ching-Hsien Hsu, Chia-Wei Chu and Chih-Hsun Chou. " 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications"

[9] Haitao Chen, Zhenghu Gong and Zunguo Huang. "Parallel and Distributed Computing Applications and Technologies (PDCAT'05)"

[10] Y. Higashi, S. Ata and I. Oka. "Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005"

[11] Jin Bo. "2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring"

[12] G.Narasinga Rao. "International Journal of Computer Applications (0975 – 8887) Volume 3 – No.2,June 2010"