



STORAGE AS A SERVICE (StAAS)

Group - 4

Abhi Patel (1641021)
Dhruvi Gajjar (1641038)
Saloni Chudgar (1641013)
Kishan Dhrangadhariya (1641048)

Instructor : Dr. Sanjay Chaudhary

Course : Cloud Computing

INDEX

No.	Title	Page No.
1.	ABSTRACT	3
2.	BASIC WORKING	4
3.	WORKFLOW DIAGRAM	5
4.	PLATFORM	6
6.	IMPLEMENTATION	7
7.	RESULTS	8
8.	CONCLUSION	16
9.	REFERENCES	17
10.	APPENDIX (CODES)	18

ABSTRACT

Storage as a service (SaaS) is a business model in which a company leases or rents its storage infrastructure to another company or individuals to store data. Small companies and individuals often find this to be a convenient methodology for managing backups, and providing cost savings in personnel, hardware and physical space. A company providing SaaS may be called a storage service provider (SSP). Storage as a service can also be referred to as hosted storage.

As an alternative to storing magnetic tapes offsite in a vault, IT administrators are meeting their storage and backup needs by service level agreements (SLAs) with a SaaS provider, usually on a cost-per-gigabyte-stored and cost-per-data-transferred basis. The client transfers the data meant for storage to the service provider on a set schedule over the SaaS provider's wide area network or over the Internet. The storage provider provides the client with the software required to access their stored data. Clients use the software to perform standard tasks associated with storage, including data transfers and data backups. Corrupted or lost company data can easily be restored.

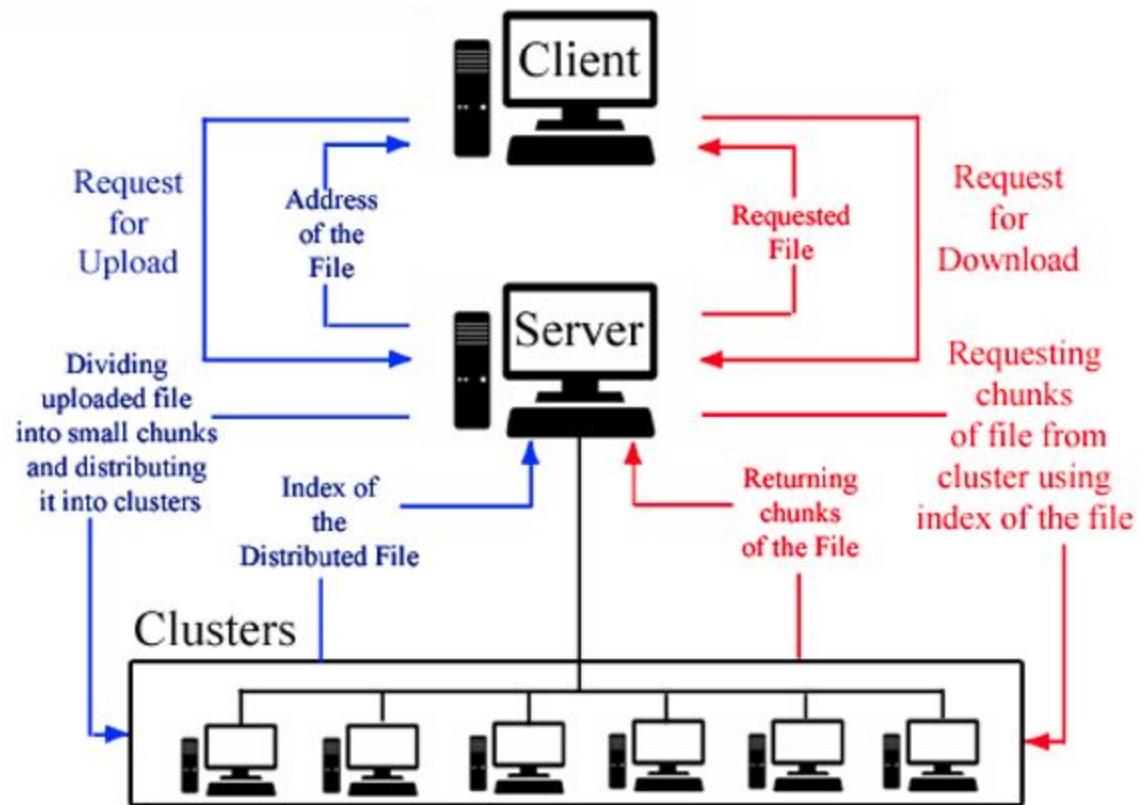
Storage as a service is prevalent among small to mid-sized businesses, as no initial budget is required to set up hard drives, servers and IT staff. SaaS is also marketed as an excellent technique to mitigate risks in disaster recovery by providing long-term data storage and enhancing business stability.

BASIC WORKING

Our project aims to build an in-house data storage solution with the complete setup in the cloud computing lab. We aim to build a cluster of computers and then setup one of them as a server computer. The user can upload a file on the web page hosted on the local network. The file will then be sent to our server computer where it will be divided into chunks. The server will then check for the available nodes i.e the number of nodes active in the cluster. The server computer will then send the chunks of the files to the active nodes. The chunks will be distributed using an algorithm which implements fault tolerance. When a file is uploaded and distributed a link is generated and given to the client. The client can then download the file using the link.

When a client requests to download a file, the request is sent to the server. The server then recombines the chunks of file from the cluster and send the file to the client.

WORKFLOW DIAGRAM



PLATFORM

Host - We used RESTful API to communicate with the client. We used Spring Framework to build the same and it was hosted using Spring Boot and Apache Maven.

Code - The code for distributing the file was written using the C language while Java and Javascript were used to implement RESTful API.

IMPLEMENTATION

Division of File:

File is divided into blocks and each block is of size of 2MB or less. Multiple copies of each block is stored as encrypted binary file with block_id and name of the file as name, in multiple physical system. Index of where the blocks are stored is maintained in server PC.

Combination of File:

Based on the information retrieved from index file created while file division, blocks are called from multiple cluster PC and are merged as a file.

Creating a Cluster:

Each node in the cluster runs the program as multi-threaded program where each thread lies on different nodes.

Implementing division and combination of files in cluster:

As an example of thread, any thread create or store a file, it will be created or stored on that particular node. So using MPI, name, size and its content is send between threads and are stored. Similarly for retrieving, each thread read and send the blocks to the main server which is asked. An index is managed in the main server which stores the details of blocks and its storage location.

Build a platform to receive and send data:

Build a web page using Java, JS and HTML, that can send and receive data over network using RESTFull-API.

Integration:

File is taken from the user through hosted page. It is then sent to program which divided files into pieces and stores it in different cluster PC, a link is provided to the user through they can download the said file.

Whenever a user clicks the link to download the file. Process of combining the file is executed and then transferred to them via network.

RESULTS

Initial Launch :

```
[File Edit View Search Terminal Help] cluster@abc-hp:~/mpich1/examples
```

```
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <!-- spring-boot-nexus-plugin:2.0.0.RELEASE:run (default-cli) < test-compile @ file-demo -->
[INFO]
[INFO] --- spring-boot-nexus-plugin:2.0.0.RELEASE:run (default-cli) @ file-demo ---

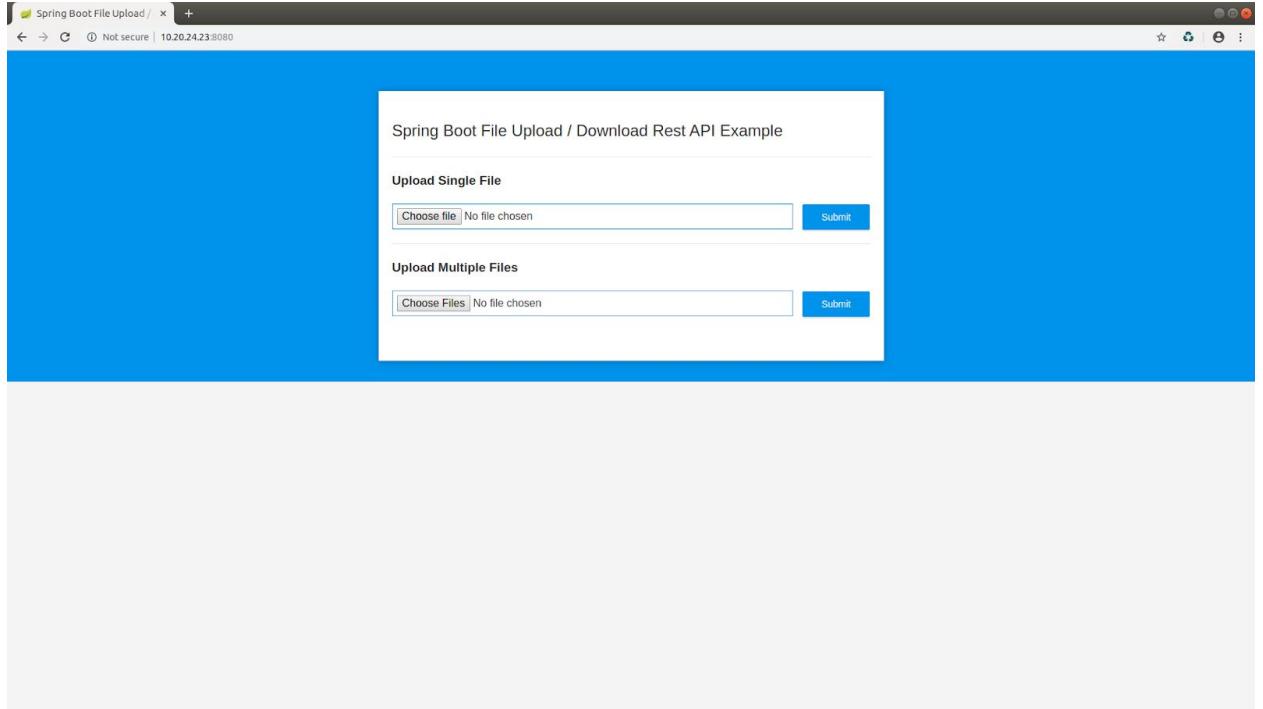


```
:: Spring Boot :: (v2.0.0.RELEASE)
```



```
2019-12-11 16:12:43.614 [INFO] 27229 --- [main] c.example.filudem.FileDemoApplication : Starting FileDemoApplication on abc-hp with PID 27229 (/home/cluster/mpich1/examples/target/classes started by cluster in /home/cluster/mpich1/examples)
2019-12-11 16:12:43.617 [INFO] 27229 --- [main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext
ext@2561a193: startup date [Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy
2019-12-11 16:12:43.617 [INFO] 27229 --- [main] o.s.b.w.embedded.tomcat.TomcatWebContainer : Tomcat initialized with port(s): 8080 (http)
2019-12-11 16:12:44.579 [INFO] 27229 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-12-11 16:12:44.581 [INFO] 27229 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.28
2019-12-11 16:12:44.594 [INFO] 27229 --- [ost-startStop-1] org.apache.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: [/home/cluster/mpich1/lib;/usr/java/packages/lib;/usr/lib/x86_64-linux-gnu/jni;/lib/x86_64-linux-gnu:/usr/lib/x86_64-linux-gnu;/usr/lib/jni;/lib:/usr/lib]
2019-12-11 16:12:44.659 [INFO] 27229 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-12-11 16:12:44.666 [INFO] 27229 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 995 ms
2019-12-11 16:12:44.784 [INFO] 27229 --- [ost-startStop-1] o.s.b.w.servlet.RegistrationBean : Servlet dispatcherServlet mapped to [/]
2019-12-11 16:12:44.784 [INFO] 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to:,['/*']
2019-12-11 16:12:44.785 [INFO] 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to:,['/*']
2019-12-11 16:12:44.785 [INFO] 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpMethodFilter' to:,['/*']
2019-12-11 16:12:44.785 [INFO] 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to:,['/*']
2019-12-11 16:12:45.101 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@context2561a193: startup date [Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy
2019-12-11 16:12:45.289 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped [[/download/file/{fileName:.+}],methods=[GET]] onto public org.springframework.http.ResponseEntity<org.springframework.core.io.Resource> com.example.filudem.controller.FileController.downloadFile(java.lang.String,java.servlet.http.HttpServletRequest)
2019-12-11 16:12:45.290 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped [[/uploadfile]] onto public com.example.filudem.payload.UploadFileResponse com.example.filudem.controller.FileController.uploadFile(org.springframework.web.multipart.MultipartFile)
2019-12-11 16:12:45.210 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped [[/uploadmultiplefiles],methods=[POST]] onto public java.util.List<com.example.filudem.payload.UploadFileResponse> com.example.filudem.controller.FileController.uploadMultipleFiles(org.springframework.web.multipart.MultipartFile[])
2019-12-11 16:12:45.216 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped [[/error]] onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,java.lang.Object>> org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.error(javax.servlet.http.HttpServletResponse)
2019-12-11 16:12:45.217 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped [[/error],produces=[text/html]] onto public org.springframework.web.ModelAndView org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.errorHtml(javax.servlet.http.HttpServletResponse)
2019-12-11 16:12:45.248 [INFO] 27229 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2019-12-11 16:12:45.248 [INFO] 27229 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.ResourceHttpRequestHandler]
2019-12-11 16:12:45.285 [INFO] 27229 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/favicon.ico] onto handler of type [class org.springframework.web.servlet.ResourceHttpRequestHandler]
2019-12-11 16:12:45.284 [INFO] 27229 --- [main] o.s.b.w.embedded.tomcat.TomcatWebContainer : Adding welcome page: class path resource [static/index.html]
2019-12-11 16:12:45.284 [INFO] 27229 --- [main] o.s.j.e.a.AnnotationBeanExporter : Registering beans for JMX exposure on startup
2019-12-11 16:12:45.424 [INFO] 27229 --- [main] o.s.b.w.embedded.tomcat.TomcatWebContainer : Tomcat started on port(s): 8080 (http) with context path ''
2019-12-11 16:12:45.429 [INFO] 27229 --- [main] c.example.filudem.FileDemoApplication : Started FileDemoApplication in 2.14 seconds (JVM running for 4.84)
```

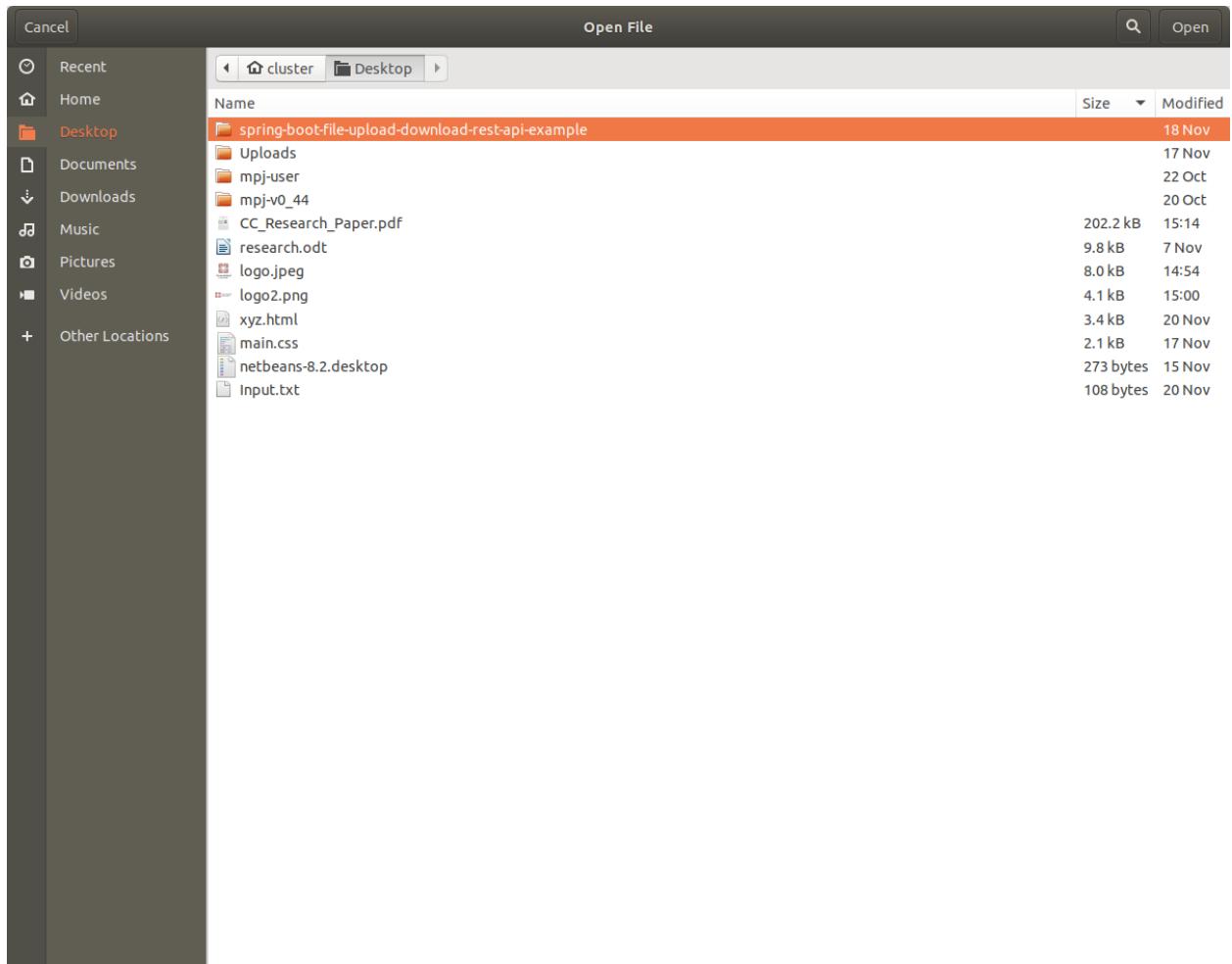

```

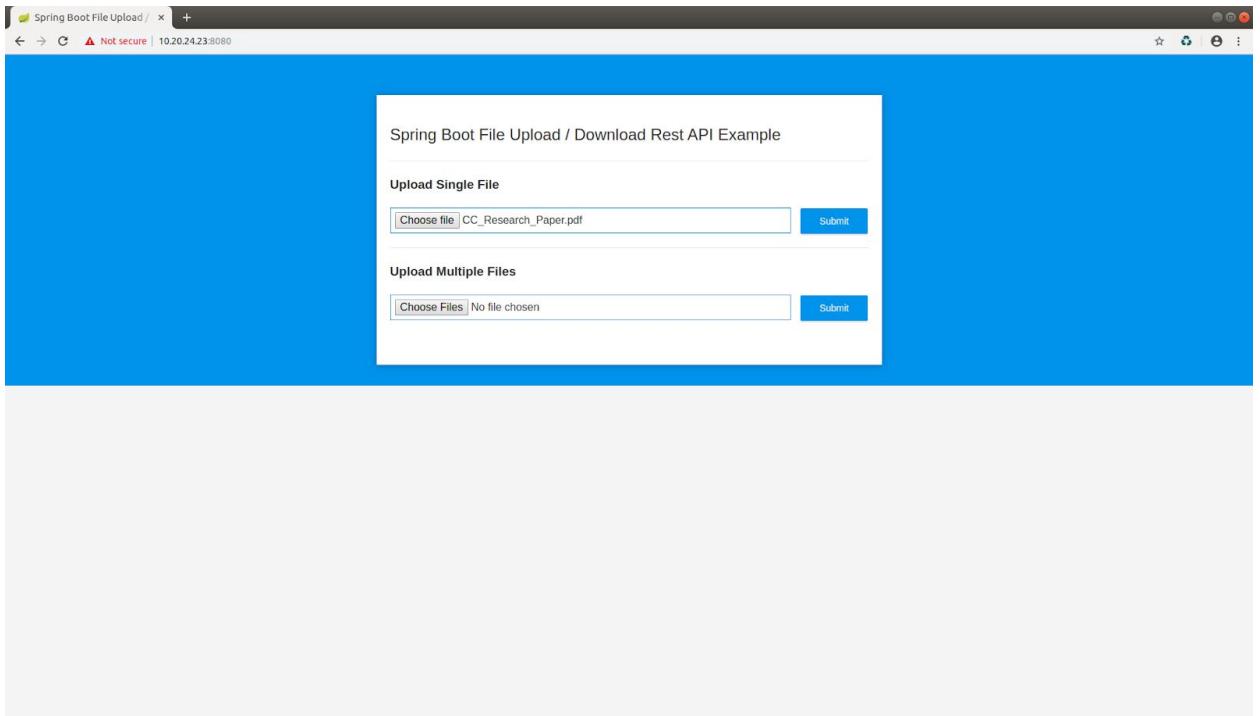


File Upload :

```
File Edit View Search Terminal Help
cluster@abc-hp:~/mpich1/examples
[1]:~$ ./Spring Boot !
2019-12-11 16:12:43.614 INFO 27229 --- [           main] c.example.filddemo.FileDemoApplication : Starting FileDemoApplication on abc-hp with PID 27229 (/home/cluster/mpich1/examples/target/classes started by cluster in /home/cluster/mpich1/examples)
2019-12-11 16:12:43.617 INFO 27229 --- [           main] c.example.filddemo.FileDemoApplication : No active profile set, falling back to default profiles: default
2019-12-11 16:12:43.617 INFO 27229 --- [           main] o.s.w.s.DispatcherServlet : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@2261aa93: startup date [Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy
2019-12-11 16:12:44.555 INFO 27229 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started with port(s): 8080 (http)
2019-12-11 16:12:44.579 INFO 27229 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.28
2019-12-11 16:12:44.598 INFO 27229 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: [/home/cluster/mpich1/lib:/usr/java/packages/lib/tls:/usr/lib/x86_64-linux-gnu/jni:/lib/x86_64-linux-gnu:/usr/lib/x86_64-linux-gnu:/usr/lib/jni:/lib]
2019-12-11 16:12:44.600 INFO 27229 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : Root Web application context: initialization completed in 995 ms
2019-12-11 16:12:44.781 INFO 27229 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2019-12-11 16:12:44.784 INFO 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [//*]
2019-12-11 16:12:44.785 INFO 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [//*]
2019-12-11 16:12:44.788 INFO 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [//*]
2019-12-11 16:12:44.788 INFO 27229 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [//*]
2019-12-11 16:12:44.789 INFO 27229 --- [ost-startStop-1] o.s.w.s.m.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@2261aa93: startup date [Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy
2019-12-11 16:12:45.269 INFO 27229 --- [           main] o.s.w.s.m.RequestMappingHandlerMapping : Mapped [[/download/file/{fileName: +}], methods=[{GET}]] onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,java.util.List<com.example.filddemo.payload.UploadFileResponse>> org.springframework.core.io.Resource com.example.filddemo.controller.FileController.downloadFile(java.lang.String,java.servlet.http.HttpServletRequest)
2019-12-11 16:12:45.281 INFO 27229 --- [           main] o.s.w.s.m.RequestMappingHandlerMapping : Mapped [[/uploadfile],methods=[{POST}]] onto public com.example.filddemo.payload.UploadFileResponse com.example.filddemo.controller.FileController.uploadFile(org.springframework.web.multipart.MultipartFile)
2019-12-11 16:12:45.281 INFO 27229 --- [           main] o.s.w.s.m.RequestMappingHandlerMapping : Mapped [[/uploadmultiplefiles],methods=[{POST}]] onto public java.util.List<com.example.filddemo.payload.UploadFileResponse> org.springframework.core.io.Resource com.example.filddemo.controller.FileController.uploadMultipleFiles(org.springframework.web.multipart.MultipartFile)
2019-12-11 16:12:45.286 INFO 27229 --- [           main] o.s.w.s.m.RequestMappingHandlerMapping : Mapped [[/error]], produces=[{text/html}]] onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,java.lang.Object>> org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.error(javax.servlet.http.HttpServletRequest)
2019-12-11 16:12:45.287 INFO 27229 --- [           main] o.s.w.s.m.RequestMappingHandlerMapping : Mapped [[/error],produces=[text/html]] onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest)
2019-12-11 16:12:45.248 INFO 27229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2019-12-11 16:12:45.248 INFO 27229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2019-12-11 16:12:45.285 INFO 27229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2019-12-11 16:12:45.304 INFO 27229 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
2019-12-11 16:12:45.384 INFO 27229 --- [           main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2019-12-11 16:12:45.424 INFO 27229 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-12-11 16:12:45.424 INFO 27229 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Using connector [TomcatConnectorFacade@127.0.0.1:8080]
2019-12-11 16:14:09.557 INFO 27229 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[]/ :
2019-12-11 16:14:09.557 INFO 27229 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Spring FrameworkServlet 'dispatcherServlet'
2019-12-11 16:14:09.567 INFO 27229 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
2019-12-11 16:14:09.567 INFO 27229 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 10 ms
written
active_pc count2
to
wall clock time = 0.839662
file distributed
file removed file/CC_Research_Paper.pdf
returned from c

```

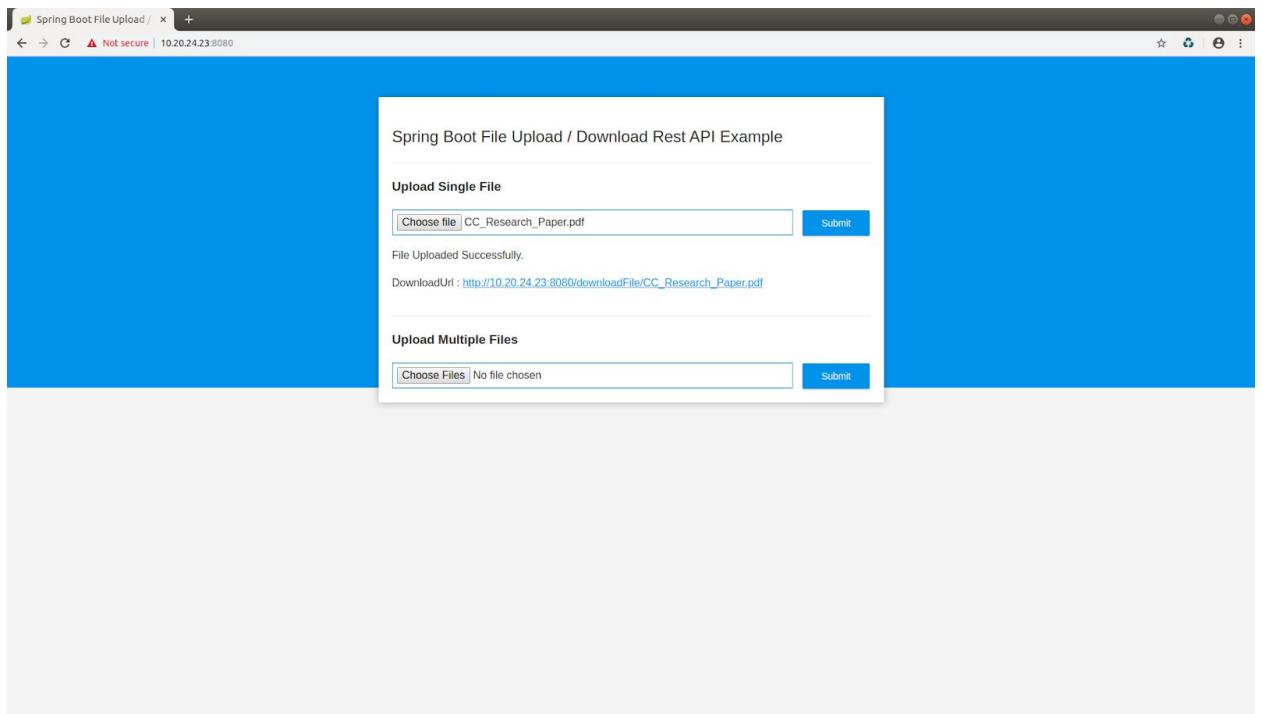




File Distribution

```
cluster@abc-hp:~/mpich1/examples/data$ ls
CC_Research_Paper.pdf-00000 examples.desktop-00000 joker.jpg-00000 'Tum HL Ho.mp3-00002' 'Tum HL Ho.mp3-index'
CC_Research_Paper.pdf-index examples.desktop-index joker.jpg-index 'Tum HL Ho.mp3-00004'
cluster@abc-hp:~/mpich1/examples/data$
```

File Download



```
cluster@abc-hp:~/mpich1/examples
```

File Edit View Search Terminal Help

tered By cluster In /home/cluster/mpich1/examples)

2019-12-11 16:12:43.617 [INFO] 27229 --- [main] c.example.filedomo.FileDemoApplication : No active profile set, falling back to default profiles: default

2019-12-11 16:12:43.617 [INFO] 27229 --- [main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext

ext@2561a193: startup date[Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy

2019-12-11 16:12:44.555 [INFO] 27229 --- [main] o.b.w.embedded.tomcat.TomcatServer : Tomcat initialized with port(s): 8080 (http)

2019-12-11 16:12:44.579 [INFO] 27229 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2019-12-11 16:12:44.579 [INFO] 27229 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.28

2019-12-11 16:12:44.598 [INFO] 27229 --- [/ost-startStop-1] o.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: [/home/cluster/mpich1/lib:/usr/java/packages/lib:/usr/lib/x86_64-linux-gnu/jni:/lib/x86_64-linux-gnu:/usr/lib/x86_64-linux-gnu:/usr/lib/jni:/lib:/usr/lib]

2019-12-11 16:12:44.660 [INFO] 27229 --- [/ost-startStop-1] o.catalina.core.AprLifecycleListener : Root WebApplicationContext: initialization completed in 995 ms

2019-12-11 16:12:44.781 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/*]

2019-12-11 16:12:44.784 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/*]

2019-12-11 16:12:44.785 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]

2019-12-11 16:12:44.785 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/*]

2019-12-11 16:12:44.785 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/*]

2019-12-11 16:12:44.785 [INFO] 27229 --- [/ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Registering for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@2561a193: startup date[Wed Dec 11 16:12:43 IST 2019]; root of context hierarchy

2019-12-11 16:12:45.209 [INFO] 27229 --- [main] s.w.s.m.n.a.RequestMappingHandlerMapping : Mapped [{@downloadToFile(fileName: +)}, methods={GET}]] onto public org.springframework.http.ResponseEntity<org.springframework.web.multipart.MultipartFile> org.springframework.web.bind.annotation.PostMapping({@downloadToFile})

2019-12-11 16:12:45.210 [INFO] 27229 --- [main] s.w.s.m.n.a.RequestMappingHandlerMapping : Mapped [{@uploadFile}] onto public com.example.filedomo.payload.UploadfileResponse com.example.filedomo.controller.FileController.uploadFile(org.springframework.web.multipart.MultipartFile)

2019-12-11 16:12:45.210 [INFO] 27229 --- [main] s.w.s.m.n.a.RequestMappingHandlerMapping : Mapped [{@uploadMultipleFiles}] onto public java.util.List<com.example.filedomo.payload.UploadfileResponse> org.springframework.web.bind.annotation.PostMapping({@uploadMultipleFiles})

2019-12-11 16:12:45.216 [INFO] 27229 --- [main] s.w.s.m.n.a.RequestMappingHandlerMapping : Mapped [{@error}] onto public org.springframework.http.HttpEntity<java.lang.String> org.springframework.boot.autoconfigure.web.error.ErrorController.error(javax.servlet.http.HttpServletRequest)

2019-12-11 16:12:45.217 [INFO] 27229 --- [main] s.w.s.m.n.a.RequestMappingHandlerMapping : Mapped [{@error}.produces=[text/html]] onto public org.springframework.web.servlet.ModelAndView org.springframework.web.bind.support.WebErrorModel.errorHtml(javax.servlet.http.HttpServletRequest)

urceptRequestHandler]

2019-12-11 16:12:45.248 [INFO] 27229 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]

2019-12-11 16:12:45.285 [INFO] 27229 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]

2019-12-11 16:12:45.304 [INFO] 27229 --- [main] o.s.b.a.w.s.HomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]

2019-12-11 16:12:45.384 [INFO] 27229 --- [main] o.s.j.e.AnnotationBeanExporter : Registering beans for JMX exposure on startup

2019-12-11 16:12:45.422 [INFO] 27229 --- [main] o.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''

2019-12-11 16:14:09.557 [INFO] 27229 --- [nio-8080-exec-1] o.c.c.c.Tomcat : [localhost/]::

2019-12-11 16:14:09.557 [INFO] 27229 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started

2019-12-11 16:14:09.567 [INFO] 27229 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 10 ms

written

active pc count

total

full clock time = 0.039062

file distributed

file removed file/CC_Research_Paper.pdf

returned from c

upload check remaining

active pc count2

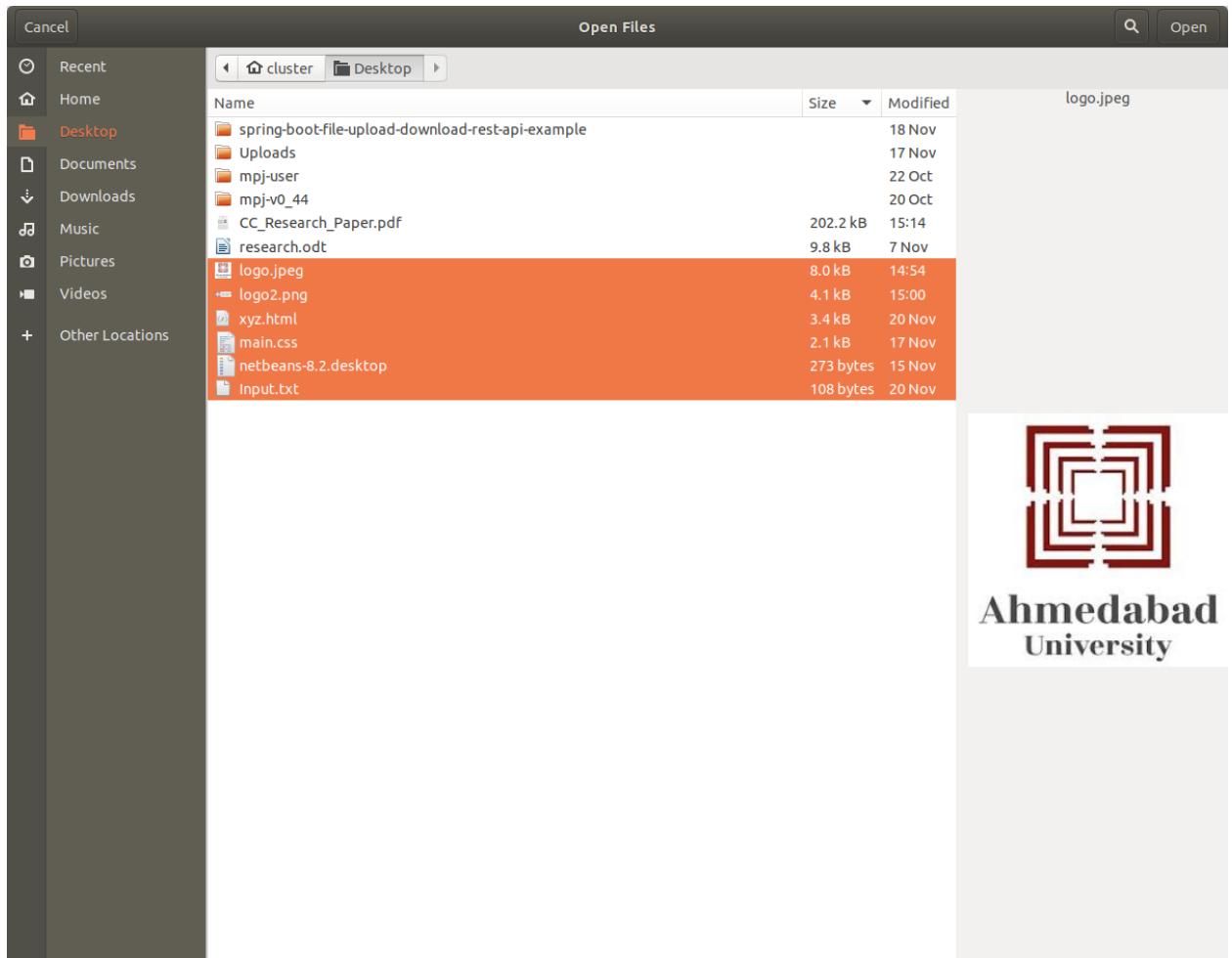
total

full clock time = 0.003906

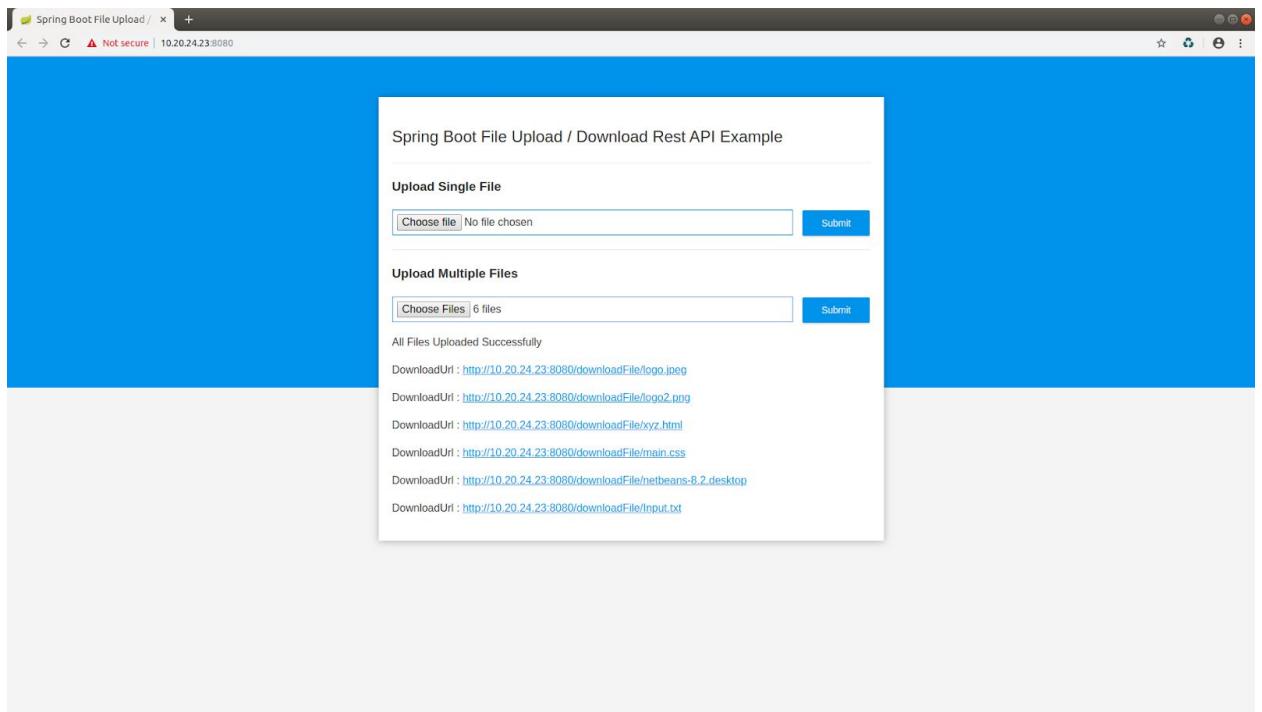
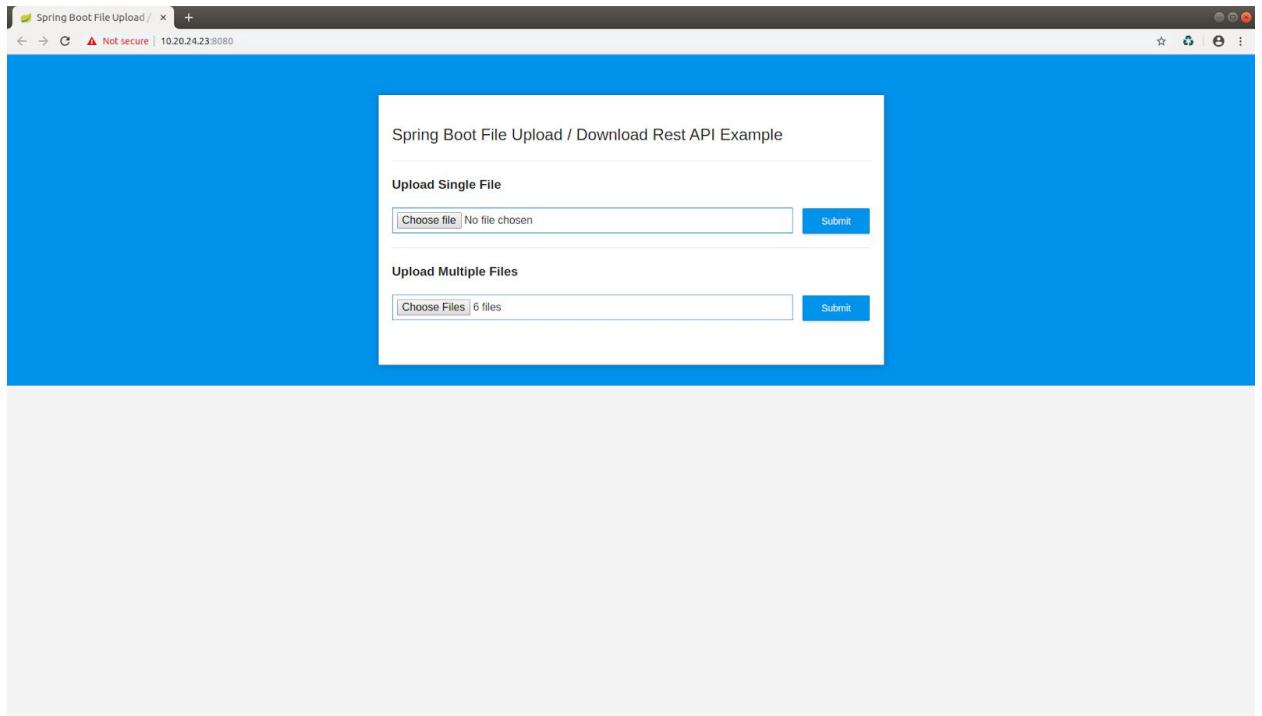
file created as file/CC_Research_Paper.pdf

returned from c

Multiple File Upload



Ahmedabad
University



```
cluster@abc-hp:~/mpich1/examples
```

```
File Edit View Search Terminal Help
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:99) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:496) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:88) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:883) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:798) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1459) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628) -[na:na]
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628) -[na:na]
at org.apache.tomcat.util.threads.TaskThreadWrappingRunnable.run(TaskThread.java:61) -[tomcat-embed-core-8.5.28.jar:8.5.28]
at java.base/java.lang.Thread.run(Thread.java:834) -[na:na]
```

```
written
active pc count2
to
wall clock time = 0.000000
file distributed
file removed file/logo.jpeg
returned from c
written
active pc count1
to
p0_31539: p4_error: : 8262
returned from c
written
active pc count1
to c
p0_31594: p4_error: : 8262
returned from c
written
active pc count2
to c
wall clock time = 0.0003906
file distributed
file removed file/main.css
returned from c
written
active pc count2
to c
wall clock time = 0.000000
file distributed
file removed file/netbeans-8.2.desktop
returned from c
written
active pc count1
to c
p0_31799: p4_error: : 8262
returned from c
```

CONCLUSION

We set up a cluster of 4 PC's in the cloud computing lab using message passing interface developed by MPICH in C language through which the files are distributed and then re-combined.

We then used spring framework and hosted it using Spring boot and Apache Maven. To communicate with the client we used RESTful API.

REFERENCES

Bouteiller, Aurelien, et al. "MPICH-V project: A multiprotocol automatic fault-tolerant MPI." *The International Journal of High Performance Computing Applications* 20.3 (2006): 319-333.

Zhang, Jing, et al. "A parallel k-means clustering algorithm with mpi." 2011 Fourth International Symposium on Parallel Architectures, Algorithms and Programming. IEEE, 2011.

Bell, William H., et al. "Evaluation of an economy-based file replication strategy for a data grid." *CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings..* IEEE, 2003.

Barron, Edward, Loellyn Cassell, and John DeGraaf. "Cluster view for storage devices." U.S. Patent No. 8,700,875. 15 Apr. 2014.

Perez, Jose Maria, et al. "Data allocation and load balancing for heterogeneous cluster storage systems." *CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings..* IEEE, 2003.

Marinkovic, Vladan Z., et al. "Load sharing cluster file systems." U.S. Patent No. 8,180,747. 15 May 2012.

Webb, Phillip, et al. "Spring boot reference guide." *Part IV. Spring Boot features* 24 (2013).

Walls, Craig. *Spring Boot in action*. Manning Publications, 2016.

Miller, Frederic P., Agnes F. Vandome, and John McBrewster. "Apache Maven." (2010).

APPENDIX

1. send.c

```
#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"
#include <string.h>
#include <time.h>
#include<fcntl.h>
#include<unistd.h>

int send0(int Ncluster, MPI_Request request, MPI_Status status, int*
cnt,char* file) {
    FILE *fp;
    int fp1;
    char filename[100];
    double startwtime = 0.0, endwtime;
    srand(time(0));
    char str[1024];
    startwtime = PMPI_Wtime();
    int i,tmp1,tmp2;
    long int val=0;

    strcpy(filename,"file/");
    strcat(filename,file);
    fp = fopen(filename,"rb");
    fseek(fp, 0, SEEK_END);
    long int size = ftell(fp);
    fseek(fp, 0, SEEK_SET);
    long int blockNo=0;
    long int size1;
    char c[6];
    char ch[100];
    val=1;
    char temp[Ncluster];
    temp[0] = 'a';
    for(i=1;i<Ncluster;i++) {
        MPI_Irecv(&temp[i],1,MPI_BYTE,i,cnt[i],MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt[i]++;
    }
}
```

```

}

strcpy(filename,"data/");
strcat(filename,file);
strcat(filename,"-index");

fp1 = creat(filename,0667);

strcpy(filename,"data/");
strcat(filename,file);
while(size>0) {
    if(size>2097152)
        size1=2097152;
    else
        size1=size;
    strcpy(str,filename);
    sprintf(c, "%05ld", blockNo);
    c[5] = '\0';
    strcpy(ch, str);
    strcat(ch, "-");
    tmp1 = rand()%Ncluster;
    tmp2 = rand()%Ncluster;
    if(tmp1==tmp2)
        if(tmp1<Ncluster-1)
            tmp2 = Ncluster-1;
        else
            tmp2 = 1;
    write(fp1, c, sizeof(char)*5);
    write(fp1, &temp[tmp1], sizeof(char));
    write(fp1, &temp[tmp2], sizeof(char));
    strcat(ch,c);
    if(tmp1!=0 && tmp2!=0) {

MPI_Isend(&val,1,MPI_LONG,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;
}

MPI_Isend(&val,1,MPI_LONG,tmp2,cnt[tmp2],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp2]++;

MPI_Isend(ch,1024,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);

```

```

    MPI_Wait(&request,&status);
    cnt[tmp1]++;

MPI_Isend(ch,1024,MPI_BYTE,tmp2,cnt[tmp2],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp2]++;

MPI_Isend(&size1,1,MPI_LONG,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

MPI_Isend(&size1,1,MPI_LONG,tmp2,cnt[tmp2],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp2]++;

while(size1>1024) {
    fread(str,sizeof(str),1,fp);
    size1-=1024;

MPI_Isend(str,1024,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

MPI_Isend(str,1024,MPI_BYTE,tmp2,cnt[tmp2],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp2]++;
}

fread(str,size1,1,fp);

MPI_Isend(str,size1,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

MPI_Isend(str,size1,MPI_BYTE,tmp2,cnt[tmp2],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp2]++;
}

else {
FILE *fp2;
if(tmp2==0);
}

```

```

    else
        tmp1=tmp2;

MPI_Isend(&val,1,MPI_LONG,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

MPI_Isend(ch,1024,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;
    fp2 = fopen(ch, "wb");

MPI_Isend(&size1,1,MPI_LONG,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

    while(size1>1024) {
        fread(str,sizeof(str),1,fp);
        fwrite(str,sizeof(str),1,fp2);
        size1-=1024;

MPI_Isend(str,1024,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;
}

fread(str,size1,1,fp);
fwrite(str,size1,1,fp2);
fclose(fp2);

MPI_Isend(str,size1,MPI_BYTE,tmp1,cnt[tmp1],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[tmp1]++;

}

size-=2097152;
blockNo++;
}
fclose(fp);

```

```

close(fp1);
val=0;
for(i=1;i<Ncluster;i++) {
    MPI_Isend(&val,1,MPI_LONG,i,cnt[i],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[i]++;
}
endwtime = PMPI_Wtime();
printf("wall clock time = %f\n",endwtime-startwtime);
return 0;
}

int send1(int rank, int Ncluster, MPI_Request request, MPI_Status
status,char * processor_name) {
FILE *fp;
int val,cnt=0;
long int size;
long int i=1;
char str[1024];
char temp;
if(strcmp(processor_name,"aryastark")==0)
    temp = 'k';
else if(strcmp(processor_name,"saloni")==0)
    temp = 's';
else if(strcmp(processor_name,"dhruvi")==0)
    temp = 'd';

MPI_Isend(&temp,1,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
MPI_Wait(&request,&status);
cnt++;

MPI_Irecv(&i,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
MPI_Wait(&request,&status);
cnt++;
while(i!=0) {
    MPI_Irecv(str,1024,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
    fp = fopen(str,"wb");
    MPI_Irecv(&size,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
}

```

```

    while(size>1024) {
        MPI_Irecv(str,1024,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt++;
        fwrite(str,sizeof(str),1,fp);
        size-=1024;
    }
    MPI_Irecv(str,size,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
    fwrite(str,size,1,fp);
    fclose(fp);
    MPI_Irecv(&i,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
}
return 0;
}

int main(int argc, char *argv[]) {
int rank,namelen,Ncluster;
char processor_name[MPI_MAX_PROCESSOR_NAME];
char file[100];
char filename[100];

PMPI_Init(&argc,&argv);
PMPI_Comm_size(MPI_COMM_WORLD,&Ncluster);
PMPI_Comm_rank(MPI_COMM_WORLD,&rank);
PMPI_Get_processor_name(processor_name,&namelen);
MPI_Request request;
MPI_Status status;

fprintf(stderr,"Process %d on %s\n",rank,processor_name);

if(rank==0) {
    int i=0, d;
    int fptr;
    fptr = open("output.txt",O_RDONLY);
    d = read(fptr, &file[i], sizeof(char));
    while(d>0) {
        i++;
        d = read(fptr, &file[i], sizeof(char));
}
}

```

```

    }
    file[i] = '\0';
    close(fptr);
//    strcpy(file,"rook.mkv");
    int cnt[Ncluster];
    for(i=0;i<Ncluster;i++)
        cnt[i]=0;
    send0(Ncluster, request, status, cnt, file);
    printf("file distributed\n");
    strcpy(filename,"file/");
    strcat(filename,file);
    if(remove(filename)==0)
        printf("file removed %s\n",filename);
}
else {
    send1(rank, Ncluster, request, status, processor_name);
}
PMPI_Finalize();
return 0;
}

```

2. rec.c

```

#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"
#include <string.h>
#include <time.h>
#include <fcntl.h>
#include <unistd.h>

int rec0(int Ncluster, MPI_Request request, MPI_Status status, int* cnt,
char* file) {
    int rank=0;
    FILE *fp1;
    int fptr;
    long int size;

    char filename[100];
    double startwtime = 0.0, endwtime;
    char str[1024];
    startwtime = PMPI_Wtime();

```

```

int tmp, i, d, sta=0;
char b, c;
long int val=1;

char name[Ncluster-1][20];
char temp[Ncluster];
temp[0] = 'a';
for(i=1;i<Ncluster;i++) {
    MPI_Irecv(&temp[i],1,MPI_BYTE,i,cnt[i],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[i]++;
//    printf("%c\n",temp[i]);
}
strcpy(str,"");

strcpy(filename,"data/");
strcat(filename,file);
strcat(filename,"-index");
char a[6];

//    printf("fptr = %d\n",fptr);

fptr = open(filename, O_RDONLY);

strcpy(filename,"file/");
strcat(filename,file);
fp1 = fopen(filename,"wb");

d = read(fptr, a, sizeof(char)*5);
a[5] = '\0';
d = read(fptr, &b, sizeof(char));
d = read(fptr, &c, sizeof(char));

//    printf("fptr = %d d = %d a = %s b= %c c = %c\n",fptr, d,a,b,c);

do {

//    printf("\n\n\n%s %c %c\n",a, b, c);

if(b=='a')
    tmp=0;

```

```

else if(b=='k') {
    for(i=1;i<Ncluster;i++) {
        if(temp[i]=='k') {
            tmp=i;
            break;
        }
    }
    if(i==Ncluster) {
        for(i=0;i<Ncluster;i++) {
            if(temp[i]==c) {
                tmp=i;
                break;
            }
        }
    }
}
else if(b=='s') {
    for(i=1;i<Ncluster;i++) {
        if(temp[i]=='s') {
            tmp=i;
            break;
        }
    }
    if(i==Ncluster) {
        for(i=0;i<Ncluster;i++) {
            if(temp[i]==c) {
                tmp=i;
                break;
            }
        }
    }
}
else if(b=='d') {
    for(i=1;i<Ncluster;i++) {
        if(temp[i]=='d') {
            tmp=i;
            break;
        }
    }
    if(i==Ncluster) {
        for(i=0;i<Ncluster;i++) {
            if(temp[i]==c) {

```

```

                tmp=i;
                break;
            }
        }
    }

    if(tmp!=0) {
        val=1;
//        printf("tmp %d\n",tmp);
        strcpy(filename,"data/");
        strcat(filename,file);

        strcpy(str,filename);
        char* ch = strcat(str,"-");
        ch = strcat(ch,a);
//        printf("%s\n",ch);

MPI_Isend(&val,1,MPI_LONG,tmp,cnt[tmp],MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt[tmp]++;
}

MPI_Isend(ch,1024,MPI_BYTE,tmp,cnt[tmp],MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt[tmp]++;

MPI_Irecv(&size,1,MPI_LONG,tmp,cnt[tmp],MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt[tmp]++;
        while(size>1024) {

MPI_Irecv(str,1024,MPI_BYTE,tmp,cnt[tmp],MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt[tmp]++;
        fwrite(str,sizeof(str),1,fp1);
        size-=1024;
    }

MPI_Irecv(str,size,MPI_BYTE,tmp,cnt[tmp],MPI_COMM_WORLD,&request);

```

```

        MPI_Wait(&request,&status);
        cnt[tmp]++;
        fwrite(str,size,1,fp1);

    }

else {
    FILE* fp2;
    strcpy(filename,"data/");
    strcat(filename,file);

    strcpy(str,filename);
    char* ch = strcat(str,"-");
    ch = strcat(ch,a);
//    printf("%s\n",ch);
    fp2 = fopen(ch,"rb");
    fseek(fp2, 0, SEEK_END);
    size = ftell(fp2);
    fseek(fp2, 0, SEEK_SET);
    while(size>1024) {
        fread(str,sizeof(str),1,fp2);
        fwrite(str,sizeof(str),1,fp1);
        size-=1024;
    }
    fread(str,size,1,fp2);
    fwrite(str,size,1,fp1);
    fclose(fp2);
}

d = read(fptr, a, sizeof(char)*5);
d = read(fptr, &b, sizeof(char));
d = read(fptr, &c, sizeof(char));
}while(d>0);

val=0;
for(i=1;i<ncluster;i++) {
    MPI_Isend(&val,1,MPI_LONG,i,cnt[i],MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt[i]++;
}
fclose(fp1);
close(fptr);
endwtime = PMPI_Wtime();
printf("wall clock time = %f\n", endwtime-startwtime);

```

```

    return 0;
}

int rec1(int rank, int Ncluster, MPI_Request request, MPI_Status status,
char* processor_name) {
    FILE *fp;
    char str[1024];
    int cnt=0;
    long int i;
    long int size;
    char temp;
    if(strcmp(processor_name,"aryastark")==0)
        temp = 'k';
    else if(strcmp(processor_name,"saloni")==0)
        temp = 's';
    else if(strcmp(processor_name,"dhruvi")==0)
        temp = 'd';

    MPI_Isend(&temp,1,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;

    MPI_Irecv(&i,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;

    while(i!=0) {
        MPI_Irecv(str,1024,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt++;

        fp = fopen(str,"rb");
        fseek(fp, 0, SEEK_END);
        size = ftell(fp);
        fseek(fp, 0, SEEK_SET);

        MPI_Isend(&size,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt++;
        while(size>1024) {
            fread(str,sizeof(str),1,fp);

```

```

        size-=1024;
        MPI_Isend(str,1024,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
        MPI_Wait(&request,&status);
        cnt++;
    }
    fread(str,size,1,fp);
    MPI_Isend(str,size,MPI_BYTE,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
    fclose(fp);
    MPI_Irecv(&i,1,MPI_LONG,0,cnt,MPI_COMM_WORLD,&request);
    MPI_Wait(&request,&status);
    cnt++;
}
return 0;
}

int main(int argc, char *argv[]) {
    int rank,namelen,Ncluster;
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    char file[100];

    PMPI_Init(&argc,&argv);
    PMPI_Comm_size(MPI_COMM_WORLD,&Ncluster);
    PMPI_Comm_rank(MPI_COMM_WORLD,&rank);
    PMPI_Get_processor_name(processor_name,&namelen);
    MPI_Request request;
    MPI_Status status;

    fprintf(stderr,"Process %d on %s\n",rank,processor_name);

    if(rank==0) {
        int i=0, d;
        int fptr;
        fptr = open("output.txt",O_RDONLY);
        d = read(fptr, &file[i], sizeof(char));
        while(d>0) {
            i++;
            d = read(fptr, &file[i], sizeof(char));
        }
        file[i] = '\0';
        close(fptr);
    }
}

```

```

    int cnt[Ncluster];
    // reset
    for(i=0;i<Ncluster;i++)
        cnt[i]=0;

    rec0(Ncluster, request, status, cnt, file);
    printf("file created as file/%s\n",file);
}
else {
    rec1(rank, Ncluster, request, status, processor_name);
}
PMPI_Finalize();
return 0;
}

```

3. FileDemoApplication.java

```

package com.example.filedemo;

import com.example.filedemo.property.FileStorageProperties;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.boot.context.properties.EnableConfigurationProperties;

@SpringBootApplication
@EnableConfigurationProperties({
    FileStorageProperties.class
})
public class FileDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(FileDemoApplication.class, args);
    }
}

```

4. FileController.java

```

package com.example.filedemo.controller;

import com.example.filedemo.exception.FileStorageException;
import com.example.filedemo.payload.UploadFileResponse;

```

```
import com.example.filiedemo.service.FileStorageService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.Resource;
import org.springframework.http.HttpHeaders;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import javax.servlet.http.HttpServletRequest;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;
import java.io.*;
import java.util.*;
import java.lang.*;

@RestController
public class FileController {

    private static final Logger logger =
    LoggerFactory.getLogger(FileController.class);

    @Autowired
    private FileStorageService fileStorageService;

    @PostMapping("/uploadFile")
    public UploadFileResponse uploadFile(@RequestParam("file")
    MultipartFile file) {
        String fileName = fileStorageService.storeFile(file);

        int temp=0;
        {
            String[] machine = { "aryastark\n", "saloni\n", "dhrushi\n" };
            try{
                Runtime aRT = Runtime.getRuntime();
                Process aProc = aRT.exec("sh check_connection.sh");
                BufferedReader reader = new BufferedReader(new
```

```

FileReader("/home/cluster/mpich1/examples/a.txt"));
    String line;
    int tmp, total_pc=3, i=0, active_pc_count=0;
    int pc_status[] = new int[total_pc];
    while((line = reader.readLine()) != null) {
        tmp = Integer.parseInt(line);
        if(tmp == 0) {
            active_pc_count++;
            pc_status[i]=1;
        }
        else {
            pc_status[i]=0;
        }
        i++;
    }
    reader.close();

    FileOutputStream outputStream = new
FileOutputStream("/home/cluster/mpich1/share/machines.LINUX");
    for(i=0;i<total_pc;i++) {
        if(pc_status[i] == 1) {
            outputStream.write(machine[i].getBytes());
        }
    }
    outputStream.close();
    temp = active_pc_count;
} catch (IOException e) {
    System.out.println(e);
};
System.out.println("active pc count"+temp);
}
Integer temp1 = new Integer(temp);
// distribute using c
try {
    FileWriter fw=new FileWriter("output.txt");

    for (int i = 0; i < fileName.length(); i++)
fw.write(fileName.charAt(i));

    fw.close();

    System.out.println("to c");
}

```

```

        Runtime aRT = Runtime.getRuntime(); //Runtime.getRuntime();
        //Spawn a shell sub-process
        Process aProc = aRT.exec("mpirun -np "+temp1.toString()+""
send.out");
        BufferedReader kbdInput = new BufferedReader(new
InputStreamReader(aProc.getInputStream()));
        String line;
        while((line = kbdInput.readLine()) !=null)
            System.out.println(line);
    } catch(IOException ex) {
    //    System.out.println(ex);
        throw new FileStorageException("Could not distribute file " +
fileName + ". Please try again!", ex);
    };
    System.out.println("returned from c");

    String fileDownloadUri =
ServletUriComponentsBuilder.fromCurrentContextPath()
        .path("/downloadFile/")
        .path(fileName)
        .toUriString();

    return new UploadFileResponse(fileName, fileDownloadUri,
        file.getContentType(), file.getSize());
}

@PostMapping("/uploadMultipleFiles")
public List<UploadFileResponse>
uploadMultipleFiles(@RequestParam("files") MultipartFile[] files) {
    return Arrays.asList(files)
        .stream()
        .map(file -> uploadFile(file))
        .collect(Collectors.toList());
}

@GetMapping("/downloadFile/{fileName:.+}")
public ResponseEntity<Resource> downloadFile(@PathVariable String
fileName, HttpServletRequest request) {

    // check if file uploaded
    System.out.println("upload check remaining");
}

```

```

int temp=0;
{
    String[] machine = { "aryastark\n", "saloni\n", "dhruvi\n" };
    try{
        Runtime aRT = Runtime.getRuntime();
        Process aProc = aRT.exec("sh check_connection.sh");
        BufferedReader reader = new BufferedReader(new
FileReader("a.txt"));
        String line;
        int tmp, total_pc=3, i=0, active_pc_count=0;
        int pc_status[]={new int[total_pc]};
        while((line = reader.readLine()) != null) {
            tmp = Integer.parseInt(line);
            if(tmp == 0) {
                active_pc_count++;
                pc_status[i]=1;
            }
            else {
                pc_status[i]=0;
            }
            i++;
        }
        reader.close();
        FileOutputStream outputStream = new
FileOutputStream("/home/cluster/mpich1/share/machines.LINUX");
        for(i=0;i<total_pc;i++) {
            if(pc_status[i] == 1)
                outputStream.write(machine[i].getBytes());
        }
        outputStream.close();
        temp = active_pc_count;
    } catch (IOException e) {
        System.out.println(e);
    };
    System.out.println("active pc count"+temp);
}
Integer temp1 = new Integer(temp);

// get from distributed file
try {
    FileWriter fw=new FileWriter("output.txt");

```

```

        for (int i = 0; i < fileName.length(); i++)
            fw.write(fileName.charAt(i));

        fw.close();
        System.out.println("to c");

        Runtime aRT = Runtime.getRuntime(); //Runtime.getRuntime();
        //Spawn a shell sub-process
        Process aProc = aRT.exec("mpirun -np "+temp1.toString()+""
rec.out");
        BufferedReader kbdInput = new BufferedReader(new
InputStreamReader(aProc.getInputStream()));
        String line;
        while((line = kbdInput.readLine()) !=null)
            System.out.println(line);
    } catch(IOException ex) {
        throw new FileStorageException("Could not retrieve file " +
fileName + ". Please try again!", ex);
    };
    System.out.println("returned from c");

    // Load file as Resource
    Resource resource = fileStorageService.loadFileAsResource(fileName);

    // Try to determine file's content type
    String contentType = null;
    try {
        contentType =
request.getServletContext().getMimeType(resource.getFile().getAbsolutePath());
    } catch (IOException ex) {
        logger.info("Could not determine file type.");
    }

    // Fallback to the default content type if type could not be
determined
    if(contentType == null) {
        contentType = "application/octet-stream";
    }
//    File file = new File("file/"+fileName);
//    if(file.delete())

```

```

        //           System.out.println("file deleted successfully");
        return ResponseEntity.ok()
            .contentType(MediaType.parseMediaType(contentType))
            .header(HttpHeaders.CONTENT_DISPOSITION, "attachment;
filename=\"" + resource.getFilename() + "\"")
            .body(resource);
    }
}

```

5. FileStorageService.java

```

package com.example.filodedemo.service;

import com.example.filodedemo.exception.FileStorageException;
import com.example.filodedemo.exception.MyFileNotFoundException;
import com.example.filodedemo.property.FileStorageProperties;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.Resource;
import org.springframework.core.io.UrlResource;
import org.springframework.stereotype.Service;
import org.springframework.util.StringUtils;
import org.springframework.web.multipart.MultipartFile;
import java.io.IOException;
import java.net.MalformedURLException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import java.io.*;

```

```

@Service
public class FileStorageService {

    private final Path fileStorageLocation;

    @Autowired
    public FileStorageService(FileStorageProperties
fileStorageProperties) {
        this.fileStorageLocation =
Paths.get(fileStorageProperties.getUploadDir())
            .toAbsolutePath().normalize();
    }
}

```

```

try {
    Files.createDirectories(this.fileStorageLocation);
} catch (Exception ex) {
    throw new FileStorageException("Could not create the directory
where the uploaded files will be stored.", ex);
}

public String storeFile(MultipartFile file) {
// Normalize file name
String fileName = StringUtils.cleanPath(file.getOriginalFilename());

try {
    // Check if the file's name contains invalid characters
    if(fileName.contains(..)) {
        throw new FileStorageException("Sorry! Filename contains
invalid path sequence " + fileName);
    }
    // if file exist
    String filename =
"/home/cluster/mpich1/examples/allfiles.txt";
    File tempFile = new File("c:/temp/temp.txt");
    boolean exists = tempFile.exists();
    if(exists==true) {
        BufferedReader in = new BufferedReader( new
FileReader(filename));
        String str;
        int tmp=0;
        while((str = in.readLine()) != null) {
            System.out.println(str);
            if(str.equals(fileName)==true) {
                tmp=1;
                break;
            }
        }
        if(tmp==1){
            throw new FileStorageException("Sorry! Filename
already exists " + fileName);
        }
        in.close();
    }
    appendStrToFile(filename, fileName);
}

```

```

        System.out.println("written ");
        // Copy file to the target location (Replacing existing file
with the same name)
        Path targetLocation =
this.fileStorageLocation.resolve(fileName);
        Files.copy(file.getInputStream(), targetLocation,
StandardCopyOption.REPLACE_EXISTING);

        return fileName;
    } catch (IOException ex) {
        throw new FileStorageException("Could not store file " +
fileName + ". Please try again!", ex);
    }
}

public Resource loadFileAsResource(String fileName) {
try {
    Path filePath =
this.fileStorageLocation.resolve(fileName).normalize();
    Resource resource = new UrlResource(filePath.toUri());
    if(resource.exists()) {
        return resource;
    } else {
        throw new MyFileNotFoundException("File not found " +
fileName);
    }
} catch (MalformedURLException ex) {
    throw new MyFileNotFoundException("File not found " + fileName,
ex);
}
}

public static void appendStrToFile(String fileName, String str) {
try {
    // Open given file in append mode.
    BufferedWriter out = new BufferedWriter( new
FileWriter(fileName, true));
    out.write(str+"\n");
    out.close();
} catch (IOException ex) {
    throw new FileStorageException("Could not store file " +

```

```
    fileName + ". Please try again!", ex);
}
}
}
```

6. FileStorageException.java

```
package com.example.filiedemo.exception;

public class FileStorageException extends RuntimeException {
    public FileStorageException(String message) {
        super(message);
    }

    public FileStorageException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

7. MyFileNotFoundException.java

```
package com.example.filiedemo.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotationResponseStatus;

@ResponseStatus(HttpStatus.NOT_FOUND)
public class MyFileNotFoundException extends RuntimeException {
    public MyFileNotFoundException(String message) {
        super(message);
    }

    public MyFileNotFoundException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

8. UploadFileResponse.java

```
package com.example.filiedemo.payload;

public class UploadFileResponse {
    private String fileName;
```

```
private String fileDownloadUri;
private String fileType;
private long size;

    public UploadFileResponse(String fileName, String fileDownloadUri,
String fileType, long size) {
        this.fileName = fileName;
        this.fileDownloadUri = fileDownloadUri;
        this.fileType = fileType;
        this.size = size;
    }

    public String getFileName() {
        return fileName;
    }

    public void setFileName(String fileName) {
        this.fileName = fileName;
    }

    public String getFileDownloadUri() {
        return fileDownloadUri;
    }

    public void setFileDownloadUri(String fileDownloadUri) {
        this.fileDownloadUri = fileDownloadUri;
    }

    public String getFileType() {
        return fileType;
    }

    public void setType(String fileType) {
        this.fileType = fileType;
    }

    public long getSize() {
        return size;
    }

    public void setSize(long size) {
        this.size = size;
    }
```

```
    }
}
```

9. FileStorageProperties.java

```
package com.example.filodedemo.property;

import org.springframework.boot.context.properties.ConfigurationProperties;

@ConfigurationProperties(prefix = "file")
public class FileStorageProperties {
    private String uploadDir;

    public String getUploadDir() {
        return uploadDir;
    }

    public void setUploadDir(String uploadDir) {
        this.uploadDir = uploadDir;
    }
}
```

10. index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0">
        <title>Spring Boot File Upload / Download Rest API Example</title>
        <link rel="stylesheet" href="/css/main.css" />
    </head>
    <body>
        <noscript>
            <h2>Sorry! Your browser doesn't support Javascript</h2>
        </noscript>
        <div class="upload-container">
            <div class="upload-header">
                <h2>Spring Boot File Upload / Download Rest API Example</h2>
            </div>
            <div class="upload-content">
                <div class="single-upload">
```

```

<h3>Upload Single File</h3>
<form id="singleUploadForm" name="singleUploadForm">
    <input id="singleFileUploadInput" type="file"
name="file" class="file-input" required />
    <button type="submit" class="primary
submit-btn">Submit</button>
</form>
<div class="upload-response">
    <div id="singleFileUploadError"></div>
    <div id="singleFileUploadSuccess"></div>
</div>
</div>
<div class="multiple-upload">
<h3>Upload Multiple Files</h3>
<form id="multipleUploadForm" name="multipleUploadForm">
    <input id="multipleFileUploadInput" type="file"
name="files" class="file-input" multiple required />
    <button type="submit" class="primary
submit-btn">Submit</button>
</form>
    <div class="upload-response">
        <div id="multipleFileUploadError"></div>
        <div id="multipleFileUploadSuccess"></div>
    </div>
    </div>
</div>
<script src="/js/main.js" ></script>
</body>
</html>

```

11. main.js

```

'use strict';

var singleUploadForm = document.querySelector('#singleUploadForm');
var singleFileUploadInput =
document.querySelector('#singleFileUploadInput');
var singleFileUploadError =
document.querySelector('#singleFileUploadError');
var singleFileUploadSuccess =
document.querySelector('#singleFileUploadSuccess');

```

```
var multipleUploadForm = document.querySelector('#multipleUploadForm');
var multipleFileUploadInput =
document.querySelector('#multipleFileUploadInput');
var multipleFileUploadError =
document.querySelector('#multipleFileUploadError');
var multipleFileUploadSuccess =
document.querySelector('#multipleFileUploadSuccess');

function uploadSingleFile(file) {
    var formData = new FormData();
    formData.append("file", file);

    var xhr = new XMLHttpRequest();
    xhr.open("POST", "/uploadFile");

    xhr.onload = function() {
        console.log(xhr.responseText);
        var response = JSON.parse(xhr.responseText);
        if(xhr.status == 200) {
            singleFileUploadError.style.display = "none";
            singleFileUploadSuccess.innerHTML = "<p>File Uploaded
Successfully.</p><p>DownloadUrl : <a href='"
+ response.fileDownloadUri +
" target='_blank'>" + response.fileDownloadUri + "</a></p>";
            singleFileUploadSuccess.style.display = "block";
        } else {
            singleFileUploadSuccess.style.display = "none";
            singleFileUploadError.innerHTML = (response &&
response.message) || "Some Error Occurred";
        }
    }

    xhr.send(formData);
}

function uploadMultipleFiles(files) {
    var formData = new FormData();
    for(var index = 0; index < files.length; index++) {
        formData.append("files", files[index]);
    }

    var xhr = new XMLHttpRequest();
```

```
        xhr.open("POST", "/uploadMultipleFiles");

        xhr.onload = function() {
            console.log(xhr.responseText);
            var response = JSON.parse(xhr.responseText);
            if(xhr.status == 200) {
                multipleFileUploadError.style.display = "none";
                var content = "<p>All Files Uploaded Successfully</p>";
                for(var i = 0; i < response.length; i++) {
                    content += "<p>DownloadUrl : <a href='"
                    response[i].fileDownloadUri + "' target='_blank'>" +
                    response[i].fileDownloadUri + "</a></p>";
                }
                multipleFileUploadSuccess.innerHTML = content;
                multipleFileUploadSuccess.style.display = "block";
            } else {
                multipleFileUploadSuccess.style.display = "none";
                multipleFileUploadError.innerHTML = (response &&
response.message) || "Some Error Occurred";
            }
        }

        xhr.send(formData);
    }

singleUploadForm.addEventListener('submit', function(event){
    var files = singleFileUploadInput.files;
    if(files.length === 0) {
        singleFileUploadError.innerHTML = "Please select a file";
        singleFileUploadError.style.display = "block";
    }
    uploadSingleFile(files[0]);
    event.preventDefault();
}, true);

multipleUploadForm.addEventListener('submit', function(event){
    var files = multipleFileUploadInput.files;
    if(files.length === 0) {
        multipleFileUploadError.innerHTML = "Please select at least one
file";
        multipleFileUploadError.style.display = "block";
    }
})
```

```
        }
        uploadMultipleFiles(files);
        event.preventDefault();
    }, true);
```

12. application.properties

```
## MULTIPART (MultipartProperties)
# Enable multipart uploads
spring.servlet.multipart.enabled=true
# Threshold after which files are written to disk.
spring.servlet.multipart.file-size-threshold=2KB
# Max file size.
spring.servlet.multipart.max-file-size=1024MB
# Max Request Size
spring.servlet.multipart.max-request-size=1050MB

## File Storage Properties
# Please change this to the path where you want the uploaded files to be
stored.
file.upload-dir=/home/cluster/mpich1/examples/file
```

13. checkConnection.sh

```
ssh -q cluster@10.20.24.12 exit
echo $? > a.txt
ssh -q cluster@10.20.24.79 exit
echo $? >> a.txt
ssh -q cluster@10.20.24.60 exit
echo $? >> a.txt
```