High Availability (HA) Distributed File Storage

Document: Project Specification

Version 1.2

Publication date: 2016-March-13

Group name: Gryffindor

Group Members:

- ARJITH CHALASANI
- JITENDRA NEELAM
- URMILA JYOTHULA
- CHINNA BALAJI YALLA
- NITISH NAGABHAIRAVA
- PRANEEL REDDY PADALA
- SURYA TEJA BOLLIMPALLI
- SRI GANESH SAI GUNNAM
- SRI KASYAP KAPPAGANTULA
- SIRISHA MANASWINI BHAMIDI
- RAGHUVINAYAK RAO MEDISETTI
- MAHAMMAD SUHAIL ATCHUKATLA

Contents:

1. Preface

The main concept of this project is to develop a secure file storage to the company SecureFile in the form of a distributed file storage system with high availability to the customers.
When a user uploads a file, the file is stored in a randomly chosen server. In this we are creating replicas for the file uploaded by the user and we use file transfer protocol for the transfer of data.

Service Developer: Gryffindor

Customer: Dragos llie

In this document we discuss about project proposal like defining technical terms, over view of customer's problems and needs, proposal solutions, limitations, time plan about how we allocated time for each stage in the project, project organisation, process tracing, quality control, risk management, system release plan and references.

- **Release v1.2 on 2016-05-15**

  Progress tracking changed
  Configuration management changed
  Risk Management changed
  Quality control changed

- **Release v1.1 on 2016-05-1**

  preface updated
  background modified
  made changes in proposal solutions
  limitations modified
  sections 8 to 13 added

- **Release v1.0 on 2016-04-18**

  Initial Release

2. Glossary and abbreviations

HTTP: Hyper Text Transfer Protocol

It is a transfer of version data formats between server and client

EX: plain txt, hyper txt, video and sound

FTPS: File Transfer Protocol Security

It is an extension for commonly used file transfer protocol(FTP) that adds support for the transfer layer security(TLS) and secure sockets layer (SSL)

Message digest: SHA-1

IT is a crypto graphic hash function which is consider practically impossible to invert that is to recreate the input data from its hash value alone.

SHA-1: secure Hash algorithm.SHA-1 produces a 160bit (20 byte) hash value known as a message digest. SHA-1 advancements are SHA-2 and SHA-3

GUI: Graphical User Interface

It is a type of interface which helps in interaction with electronic devices through graphical icon and visual indicators.

SQL Server: Structured Query Language Server

SQL is used to store, query and manipulate data. It is used for manage data in a relational data base.

Restful API: Representation State Transfer

An architectural pattern to improve probability and scalability of a system.

3. Background

SecureFile offers a secured file storage to the customers using FTPS based system but the customers of secureFile are no longer satisfied because of FTP connections, which is difficult on local machine, Servers can be imitated to send data to a random port on an unintended computer, performance problems, maintains problems and availability problems

So we are designing a new service in the form of a distributed file storage with high-availability to the SecureFile company to overcome this situation.

4. **Proposed Solution**

The proposed solution is to provide a file storage in the form of a High-Availability (HA) distributed file storage system to the customers. High availability basically means that redundancy throughout network infrastructure
To accomplish this task, first a system administrator decides the initial number of servers in the distributed system and the number of file replicas maintained for each file. When user uploads a file the information is to be stored in a randomly chosen server and it creates the replicas for the files. the server constantly checked by a pinging process. Each server regularly sends a message to each of the other servers. The recipient must acknowledge the reception of the message. If a recipient fail to do this multiple times in a row the server is marked as down. if the server is brought back a constant verification is done, if it fails in verification again it is taken as a server down case. We define a namespace on the server for each user and unique root folder is assigned to each user so that they can upload same identical files. We Provide encrypting to the files for security and also use open SSl for certifications

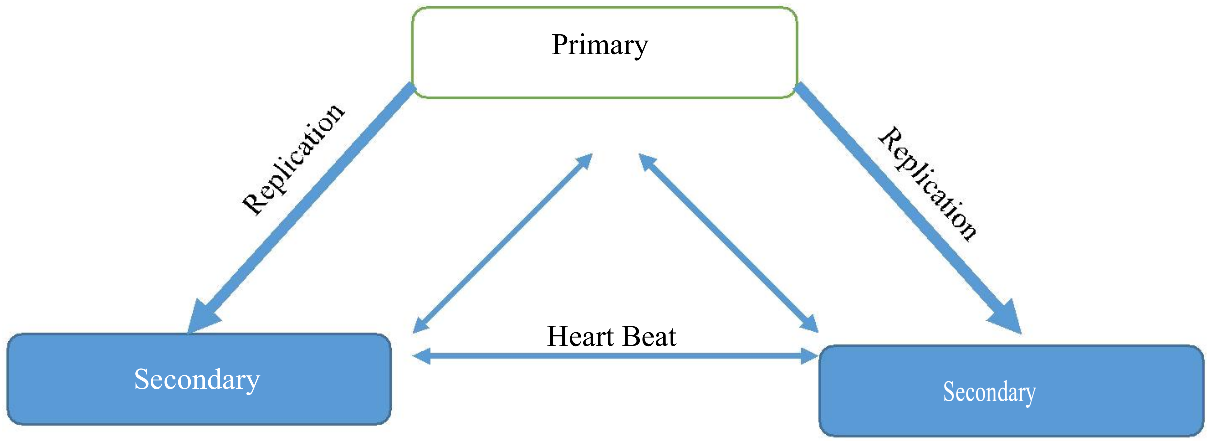

Fig 1: Heart Beat Mechanism

5. Limitations

- Maintaining consistency among copies when a replicated file is updated is a major design issue of a distributed file system that supports file replication.
- The total storage space of each user is limited to 7 GB because of the limited storage space in the server database.

6. Time Plan

Week	1	2	3	4	5	6
Starting	April (10-17)	April (18-24)	April-May (25-1)	May (2-8)	May (9-15)	May (16-22)
Project Specifications Docs	🟦					
Project Specifications Binaries	🟦					
SRS Docs		🟦				
SRS Binaries		🟦				
Design Docs			🟦			
Design Binaries			🟦			
Accept Plan Docs				🟦		
Accept Plan Binaries				🟦		
Release Candidate					🟦	
Product Release						🟦

Fig 1: Time Plan

Toll Gates:

- 2016-04-17: Project proposal
- 2016-04-24: Software Requirements Specifications
- 2016-05-08: Acceptance test plan
- 2016-05-22: Product Release

Milestone:

- 2016-04-17: Project Specification
- 2016-05-01: Design document
- 2016-05-15: Project Documentation

7. Project Organization:

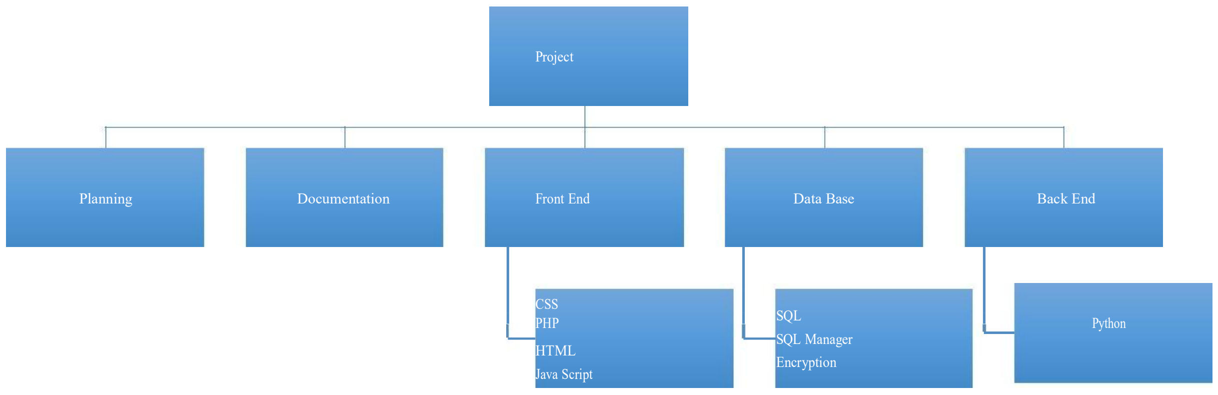

Fig1: project organization

Task Splitting:

1)	Documentation	P Praneel Reddy., G Ganesh., N Nitish., B Manaswini
2)	CSS	Ch Arjith., J Urmila
3)	PHP	K Sri Kasyap., B Surya Teja., A Mahammad Suhail., Ch Arjith.,N Nitish
4)	HTML	J Urmila., M Raghu., N Jitendra., G Ganesh
5)	Java Script	B Manaswini., M Raghu., N Jitendra
6)	SQL	P Praneel Reddy., K Sri Kasyap., Y Balaji
7)	SQL Manager	K Sri Kasyap., Y Balaji
8)	Python	A Mahammad Suhail., B Surya Teja

8. Configuration Management

VERSION MANAGEMENT:

In version management we keep track of different versions. The process of version management is done through gitlab. Git lab is used to upload all the files in the storage space and we can also update the code.

- Storage Management: - In this gitlab is present where we store the files and these files can be replaced, delete and merge. These files can be downloaded by any member in the group.

- Project Support: - The gitlab supports many files we store different files in different places in the git lab a separate place for bin where the deleted files are stored. Libraries are also present to share data. The check ins and check outs of the members can be checked over the graphs the commits committed by the members are also seen through it.

- Independent Development: - In a group the task is divided among all the members and an individual coding language is divided among different members the code developed by individual developer are uploaded into the git lab and these codes can be seen by the rest of the developers in the group if they want to develop it they can replace them.

SYSTEM DEVELOPING:

It is the process of creating a complete executable system by compiling and linking the system components external libraries etc.
System building tools and version management must communicate as the build process involves checking out component versions from the responsibility.
In system building the development system and the build server may both interact with the version management system the version management system may either be hosted on the build server or on a dedicated server.
In order to develop a proper system, we need to do the following things:

Build script generation: -
it must analyse the program that is being built identify dependent components and automatically generate a build script
Version management system integration: -
the build system should check out the required versions of components from the version management system

Reporting: -
the build system can automatically run automated tests using test automation tools such as junit these check that the build has not been broken by changes
Documentation generation: -
The build systems may be able to generate release notes about the build systems and system help pages
The build system includes code lines and base lines according to the changes done in the code lines and base lines the concluding build is checked out to build server to build definitive editions of the software and final software is compiled.

RELEASE MANAGEMENT:

The release management is categorised into major releases and minor releases. but in this project we are not providing any major releases but we provide all the details how to perform the installation process and usage process of the given product. If any minor releases are present those are included in the usage documentation.

9. <u>Progress Tracking</u>

The group meetings are conducted every week in order to describe the work that is performed by the group. Milestones are used in order to see if any work or submission has to be performed. Issue tracking n gitlab is used by the members if any issues are faced by them these issues are discussed by the group members over the meetings. The project task distribution is done through project libre and a task is given to every member and in the meetings the overall project is discussed and documentation is done for that week.

10. <u>Quality Control</u>

In this project we ensure that proper quality of the product is provided to the customer. The basic need or the aim of the product is to provide high availability we shall make sure that the system never goes down or the loss of the file never occurs in the project this can be done through the replication process.

The quality team sees that the customer requirements are satisfied. The complexity of the code is checked and if any complexity is present than we must check for the simple codes.at the end tests must be performed in order to identify the bugs in the product.

In this project even if ones’ server is down we should be able to access the files. this has to be done to replication of the files. We need to make sure that the replication does takes place in the server

QUALITY PLAN:

Product introduction a deep description of the project its intended market and the quality expectations of the project.
Product plans the critical release dates and responsibilities for the product along with the plans for distribution and product servicing.
Quality goals the quality goals and plans for the product including an identification and justification of critical product quality attributes
Quality documentation is a record of what has been done by each subgroup in this project. It helps people or the customers that the important tasks have not been forgotten or one group has not made incorrect assumptions about what other groups had done.
The quality management must answer few questions. if the project had answered these questions than the project can be released

- Has the software been properly tested?
- Is the software sufficiently dependable to be put into use?
- Is the performance of the software acceptable for normal use?
- Is the software usable?

In order to ensure the quality management software standards must be ensured for the product. There are two types of standards

Product standards: -
These apply to the software product. they include the document standards, coding standards (which define how a programming language should be used). product standards have to be designed so that they can be applied and checked in an as a cost effective.

Process standards: -
These define the processes that should be followed during software development they include specifications, design and validation processes.
Review and Inspections are QA activities they check the quality of project deliverables. This involves examining the software its documentation and records of the process to discover errors and omissions to see if the quality standards have been Followed. The Conclusion of the review should be recorded as part of the quality management processes.

11. <u>Risk Management</u>

In a group while performing a project we need to access the risks that may affect a project monitor their risks and take action when problems arise
Risk management involves anticipating risks that might affect the project schedule or the quality of the software being developed.

The strategies followed by our group are

Avoidance strategies: - we use this strategy in which we avoid the use of defective components which might lead to troubles. Requirements changes must be avoided by derive traceability information to asses' requirements change impact, maximising information hiding in the design.

Contingency plans: we follow a contingency plan in order to prepare for the worst condition and using a backup process for running the project with a simple solution with less advantages.

Minimisation strategies: this a strategy we follow when a group member is sick by distributing his work among all the team members and increasing the working hours of the rest of the team

11. system release plan:

12.1 Testing plan:

After developing the complete product, tests will be performed. In a group few people perform the tests required for the detection of the bugs and ensure efficiency of the product. The tester of the product is the developer of the software. For a formal process a separate testing group within the development team is used.

Testing is carried out in a three levels of granularity:

1. unit testing:

Where individual program units or object classes are tested.it focuses on the functionality of objects or methods.

2. Component testing:

It is used to focus on testing component interfaces. Where several individual units are integrated into a composite component

3. system testing

Where some or all the components in a system are integrated and the system is tested on a whole.

Development testing is primarily a defect testing process, where the aim of the testing is to discover bugs in the software.it is therefore usually interleaved with debugging the process of locating the problems with the code.

12.2 Packaging plan:

A compressed tar.gz file is provided to the user consisting all the source code, library files and related documents. The details of the release plan are given

Release candidate

Final release

12.3 documentation plan:

12.3.1 installation documentation:

The installation document is in the pdf format it consists all the information bout installation process for the different software's used in the product and also the configuration settings for different components

Time schedule

It is also divided into two phases they are

Documentation testing phase
Documentation release phase

12.3.2 user documentation: -

the user documentation is in the form of the pdf format in here the detail information of the product is present it contains details like installation of the product and usage of the product
Documentation before testing
Documentation after testing

12.3.3Developer documentation: -

This document gives scope to the future scope of the project it consists all the source codes and all the related information about the API'S used in the project
Documentation before testing
Documentation after test phase
Time schedule
User documentation is prepared after the installation documentation is done considering the fact that the user has no prior knowledge regarding the usage of the tool.

13. References

- Sommerville, Ian. Software Engineering, 9th ed. Addison-Wesley, 2011