GNU PROJECT - GUIX

Decentralised Substitute Distribution

Mentors:

- pukkamustard
- Attila Lendvai

VIJAYA ANAND (sunrockers8@gmail.com)

B.S., Mathematics & Scientific Computing

Indian Institute of Technology, Kanpur

Kanpur - 208016, India

Summary

This project aims to increase the robustness of GUIX substitute distribution by using a more decentralised approach. Substitutes are pre-built packages, archived into .nar files and downloaded majorly from a central server (build farm). To lessen the load of this server we could obtain these substitutes through a P2P network for eg IPFS. But instead of distributing the .nar files into the network we encode the substitutes using ERIS, which basically break up input data into uniform encrypted blocks, which are then distributed in the network directly. This ensures that the encoding of the data is independent of the transport layer used. The blocks can be accessed back only a reading capability (produced as a result of the ERIS encoding).

The project mentor, *pukkamustard,* has worked in this direction and released patches to convert obtaining substitutes from a central server (build farms) to IPFS. **The main contribution of this project would be to implement fallback mechanisms in case receiving substitutes from these P2P networks fail**. For instance, in case IPFS fails, we need to fall back to another transfer protocol or receive blocks directly from the substitute server thus giving a hassle free experience when a user wants to retrieve substitutes.

This project will include determining the point of failures which can occur when receiving the substitutes over any network and implementing the corresponding fallback mechanism. **It will particularly focus on the case of unavailability of the substitute blocks over the network.**

Benefits (Why this project is important)

Decentralised distribution of substitutes has several advantages such as :

- Increased reliability of substitute downloading by relying on a larger set of servers/peers.
- Reduced load on build farms.
- Reduced network usage by being able to use substitutes available closer (e.g. on local network)

Further, ERIS encoding the substitutes makes distribution of substitutes protocol independent. Any protocol that can transfer small blocks (size 32 KiB) referenced by the hash of the data can be used, for example IPFS, GNUNet, HTTP etc…

Previous work by *pukkamustard* has enabled distribution/fetching of ERIS encoded substitutes through IPFS network, referenced by a ERIS URN available in the narinfos of the substitute. Work has also been done for fetching ERIS encoded blocks over HTTP protocol.

With this GSoC project, I aim to implement a fallback mechanism in case receiving a substitute (as ERIS encoded blocks) over any P2P mechanism fails.
Lets say we have implemented the distribution of substitutes over a P2P network. A user wishing to receive a particular substitute should not be faced with any errors, for example missing of data blocks in the network. The user should not be burdened with the prospect of facing errors like these just because we want to implement a decentralised distribution. In order to let the user have a hassle free experience, we need to have some fallback mechanism.

This project would thus help to maintain the benefits of decentralised substitute distribution itself as well as give guix users a hassle free experience while receiving the substitutes.

Deliverables

The final deliverables would be a series of patches which would handle failures arising from a decentralised distribution of substitutes and implement a fallback mechanism to avert them. These set of patches would add functionalities to the existing patches (created by *pukkamustard*).

The broad problem statement the patches will solve:

- Determining the point of failure when receiving/publishing substitutes through some P2P network.
- Implementing the fallback mechanism to handle the failure.

Tasks to be implemented (missing data blocks in the network failure):

- We have to see how the blocks are to be stored in the substitute server. We have to find the optimal storage structure so as to retrieve missing blocks fast.
- Accessing corresponding missing data-blocks from the substitute server. This is probably the most important problem that needs addressing.
- We have to decide on a point of failure after which we start to request for blocks from substitute servers through HTTP or some other transport method.
- Making the missing blocks accessible back in the network so that this problem doesn't occur again.
- Document the work and writing tests for the above implementation.

Implementation details :

- In case a user requests for a substitute and there is a missing block in the decoding process, a HTTP request for block would sent to the substitute server and the server will encode the corresponding block in real time and push it back into the network. The block will be searched again and retrieved.

The code for the above implementation point will be for both the client side and the server side. There will be some endpoints implemented on the server side (such as for posting the missing encoded blocks back onto the network).

- Failure can be decided in case the block is not retrieved from the network until a threshold time is reached, let's say some N seconds. After this time, we would send the missing block request to the server.

It is important to document this implementation so as to explain the logic behind the same. It will also aid in receiving constructive comments from the guix community and help make necessary changes to the code.

Writing tests for the implementation is also required to test that the patches indeed work. The test environment for the implementation can be:
Locally building a test substitute and creating its narinfo with ERIS urn and then publishing only some selected ERIS encoded blocks to the IPFS network. We can then try to receive that dummy substitute while a local substitute server (lets say on a VM) acts as our fallback block provider.

Plan

I will maintain close contact my project mentors during all times, and will update them daily and if required even multiple times a day through email so that the mentor can track progress as I work on the project.
I will fork the guix source code into my Github account and make commits to the same as I code. This can also be viewed by the project mentors and they can give reviews on the same. Once significant changes have been made to the codebase and an objective of the project can be solved, the commit bunch can be sent as patches (RFC) to the community.

- **Evaluation 1 :**

  The mid term evaluation can be considered a success if there is a system in place for storage of data in the server side which would optimise serving of missing blocks to the user upon request and also the implementation of the response from the server side to request of missing blocks. I will also test and document the above implementation.

- **Evaluation 2 :**

  The end term evaluation can be considered a success if there is a system in place for the user side to receive the .nar files even in case the retrieval of blocks through the P2P network fails. I will also the test the fallback mechanism as a whole using the test environment as mentioned above in the implementation details.

Timeline

I have provided below an expected timeline I plan to stick to, so as to complete the project. Though I have planned the timeline it may be possible that there are unplanned deviations in the schedule. I plan to cover up for them by working off schedule. I will not be having any other commitments till August (after which my college semester starts) and GSoC would be my primary activity. Although I will be staying in the campus for some research work but it shouldn’t pose any issues since it would account for only couple of hrs of work a day.

Pre GSoC Period	
Present - April 15	Get more comfortable with Guile Scheme and complete reading relevant parts of the guix codebase (publish.scm/substitute.scm etc…).
April 15 - May 4	Get the patches created by *pukkamustard* working in local system.

Community Bonding Period	
May 4 - May 28	1. Clear any doubts and issues arising while running the above patches. 2. Research on the different point of failures which could need a fallback mechanism. 3. Discuss with community about the fallback mechanism to be implemented and ensure practical feasibility and logical soundness of the same. 4. Write an implementation and testing document and finalise the same after discussions.

Coding Begins!	
May 29 - July 10	1. Implementation of server side database. 2. Implementation of serving missing blocks by local server based the user request. 3. Testing above implementations and working on mentor reviews. 4. Working on any other points of failures that may arise.

Mid Evaluations (July 10 - July 14)	
July 14 - August 28	1. Implement user side request of missing substitute blocks to server when threshold for failure reached 2. Implement receiving of missing block and resuming process of deploying requested substitute. 3. Testing the entire fallback mechanism and working on mentor reviews. 4. Buffer time

Final Evaluation (Aug 21 - Aug 28)
GSoC Ends :(

Communication

As mentioned above, I will be communicating daily with my mentor through mail and will also keep making commits to the forked guix source code in my Github so that the mentor will be able to see the code as it progresses.
We could also keep weekly online meets every weekend so as to discuss my progress, clear doubts and receive reviews from the mentors.

Qualifications

Motivation for choosing this project

Nearly completing two years of undergraduate coursework at IIT Kanpur, I have been constantly developing interest in Cryptography and Distributed Systems and have done a lot of self reading on these topics. I am also taking a cryptography course this semester in my institute. I am also very interested in protocols for decentralised file sharing and have wondered how data can be transferred without a central server. Hence naturally, I'm very interested to work on this project which is about implementing a decentralised distribution of guix substitutes.

I have an idea about how various network protocols work like HTTP, IPFS etc.. work. Although I have some idea about the theory part of these protocols (like the protocol IPFS uses), I don't have that much of an idea about how these protocols are coded. I think that this project gives me the perfect opportunity to learn more about the implementation side of a distributed data sharing protocol and how data is encoded before sharing it in this type of a protocol. Although this project does not directly involve implementing of a distributed file sharing system, researching the way the P2P network, on which we are deploying the .nar files, work is required so as to identify points of failures while retrieving the data. I am also very interested in pursuing higher studies in the field of distributed systems and cryptography and hence I will benefit from getting exposure to the real life implementation of a related system.

After GSoC

I wish to continue working on this topic of decentralised guix substitute distribution even after the GSoC period. Just providing of a fallback mechanism will not be sufficient for this idea to achieve full fruition. There are many more factors to be considered and networks to be explored, which can be used for the sharing of ERIS encoded blocks. If this type of ERIS encoded substitutes data block sharing is implemented in many other P2P networks, an extra fallback mechanism which searches for blocks in other P2P networks can also be investigated.

There is also the matter of investigating which compression method (acting on the nar file) will be most ideal in terms of de-duplication of data blocks in the substitutes. A file system like EROFS which compresses the given files into uniformly sized blocks could be more efficient in terms of de-duplication as ERIS itself encodes data block wise. There has been research going on regarding this and an ERIS File System has been proposed for efficient file de-deduplication while encoding data with ERIS. I find this topic area very I would love to work on it after the GSoC period ends.

Background

I have very recently started to explore and contribute to free software and would love to utilise this GSoC as the entry point to working on free software. I have experience with kernel development and have submitted some dt-binding patches to linux kernel.

Coming to other programming experiences, I have experience in coding with C, Python and Bash and have also done some web development using Django.

I had come into touch with *pukkamustard* a couple weeks back and they had given various reference materials required to come to pace with the theory required to work on this project. I have read almost all the material and this has helped me to understand better the problem in hand.

Skills that I have relevant to the project :

- Understanding of network protocols.
- Understanding of how ERIS encodes data.
- Understanding of the basic working of guix and how guix substitutes are packaged and distributed.
- Idea behind how guix substitutes can be distributed and received from a P2P network like IPFS based on the patches created by *pukkamustard.*
- Basic idea and syntax of Guile Scheme, which is the language in which guix is implemented.
- Using REPL and Emacs for Guile Scheme coding.

I am yet to become fully comfortable coding in Guile and using emacs for the same. I will have to learn the same so as to fully understand and run the above mentioned patches in my local system.

Contact and other details

Name : Vijaya Anand

Phone : (+91) 7550080422

Email : sunrockers8@gmail.com

Institute : Indian Institute of Technology, Kanpur

Github : ppopdesk