GSoC'23 Proposal
For
The GNU Organisation

Mentors:
pukkamustard
Attila Lendvai

Name:
Shivam Madlani

E-mail:
shivammadlani5@gmail.com

Project:
Decentralized substitute distribution

Summary:-

The aim of the project is to robustify the current system for downloading substitutes in The GUIX Package Manager. Currently the servers upload pre-built packages as a compressed archive file (.nar). Instead we could use ERIS. It breaks down content into smaller chunks (1KiB or 32KiB according to the content size) which will then be transported through the network/sneakernet. ERIS would allow multiple transport protocols like IPFS(InterPlanetary File System), HTTP and GNUnet.
My proposal is to use an external drive to transfer substitutes. This will be better for users with poor internet connectivity. The package content can be encoded using ERIS and stored in SQLite database in drives/flash drives. The portable database can then be decoded and used on any supporting computer.

Benefits:-

Implementing this project will robustify the current setup. Currently the two build farms (*ci.guix.gnu.org* and *bordeaux.guix.gnu.org*) remain the primary source for most users and also the primary points of failure.

GNU benefits:
This project will allow substitutes to be air-gappable and thus reduce load on the build farms.

User benefits:
More reliable transfers.
Significant Improvement of file transfer speeds and reduces network usage.

Deliverables:-

Library to be integrated with GUIX: guile-eris

Currently the Guile implementation of ERIS is done in guile-eris library. This can be used to encode/decode content using ERIS standards.

- About ERIS:

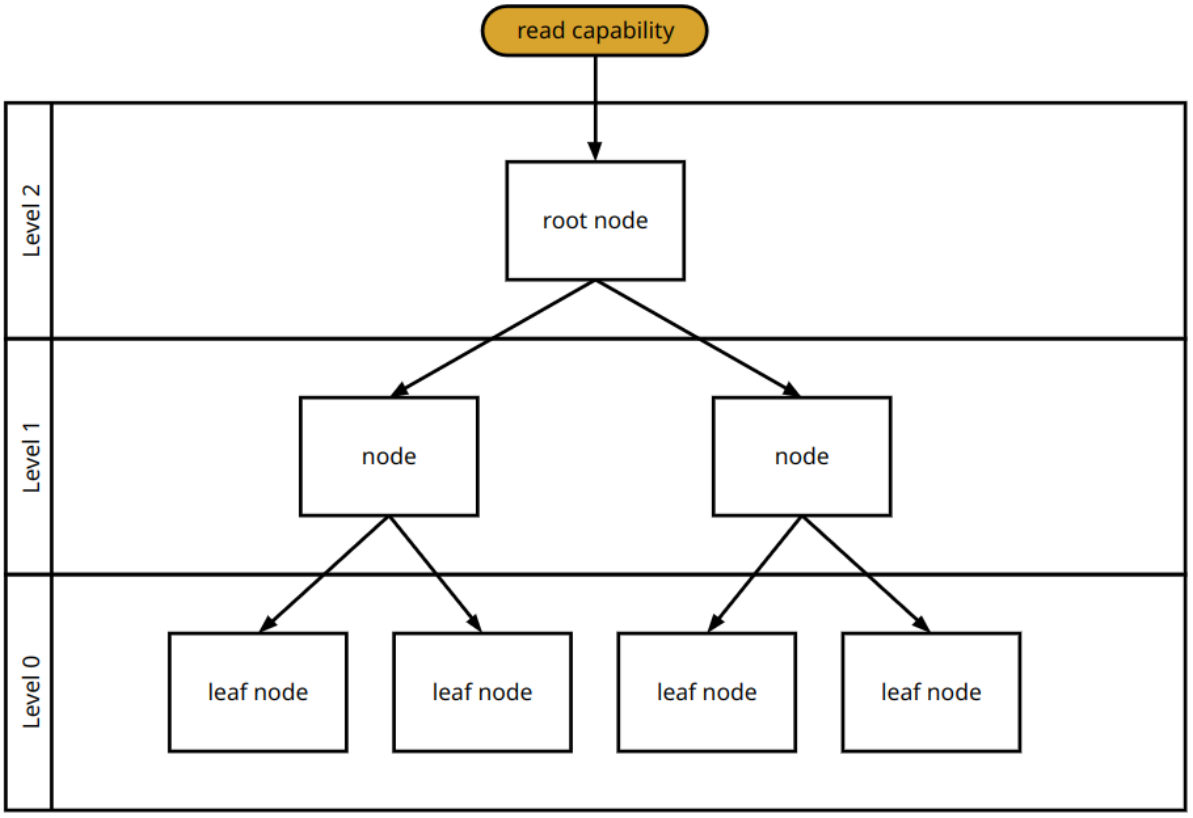
  ERIS (Encoding for Robust Immutable Storage) is a method of encoding content into uniformly sized encrypted blocks with read capability. Each block is assigned a content-id (CID) using which the content can be reassembled. The content can also be referenced with a URN.

- How ERIS encodes:

  ERIS breaks down content into uniformly sized blocks of size (1KiB or 32KiB). ERIS will automatically add padding to the blocks having data lesser than the block size.
  For content smaller than 16KiB- block size of 1KiB is recommended
  For content larger than 16KiB- block size of 32KiB is recommended for better efficiency.

  - The encoding process creates a binary tree of uniformly sized encrypted leaf nodes.
  - The leaf nodes (level 0) represent the actual encoded content.
  - The internal nodes (except head and level 0) consist of references to nodes at lower levels.
  - The root node denotes read-capability.

The below figure visualises encoding content that has 4 leaf nodes.



*reference: [ERIS specification document]

- How ERIS decodes:

  To decode content we need:

  - The reference to the root node
  - The key to decrypt the root node
  - The level of the root node

Software added/changed

We will be adding SQLite to store the substitutes.

Affected project code

To store any data in a database we have to create a schema. Similarly, in order to use SQLite we need to define a schema. Later on we can add data of the substitutes into the tables in the form of a BLOB (Binary Large Object).

We define our table like this:-

```
CREATE TABLE IF NOT EXISTS eris_blocks{
        Ref BLOB PRIMARY KEY,
        block BLOB
}
```

After defining table we can run SQL queries like:

```
INSERT INTO eris_blocks (eris-encoded-data) VALUES (XYZ)
```

Where XYZ will be ERIS encoded blocks.

In order to retrieve all the contents from the DB we use SQL query:

```
SELECT (eris-encoded-data) FROM eris_blocks WHERE id=x
```

Affected documentation:

Documentation related to SQLite will have to be added in the guix user manual in case the user wants to know more about sneakernet.

User experience with the added code:

The user running `guix install {xyz}` will be able to pass a flag (e.g. --sneakernet-write and –sneakernet-read). If the computer detects any compatible drives, the user will be asked to select it and after that the package's installation will begin. Then encoding using eris will be done by the user who writes substitutes to the disk and decoding will occur on the user installing from the drive.

Plan:-

Proposed Timeline:
The below timeline shows roughly on what I'll be working and for how long.

Pre-GSoC

Present day to 10th April	Learn more about guile-SQLite bindings and exploring guix codebase.
12th April to 4th May	Interacting with the mentors to finalise my approach and get to plan ahead for potential blockers or unknown issues that might occur in the coding period.

Community Bonding period

4th May to 28th May	Clearing any sort of doubt including the setup or the approach to the problem. Getting to know the mentors and about the time during which I can contact them. Creating a written roadmap for the project and setting up guix on another computer which will be used for testing purposes.

Coding period

29th May to 22nd June	Implementing the storing of encoded ERIS blocks into SQLite database. If this part is successfully implemented then the rest is easy. It will take a bit of time to learn about guile-sqlite bindings but once I'm familiar it'll be easy. At this point I believe the project will be 60% complete.

22nd June to 2nd July	Implement the decoding of SQLite data back to the substitutes.
2nd July to 8th July	Buffer period for unexpected bugs and errors
8th July to 10th July	Thoroughly testing my project on multiple computers(Virtual and physical).

Midterm-evaluations(10th to 14th July)

14th July to 21st August	Improving on the feedback provided by the mentors during the mid-evaluations and working on cleaning up the code. Making minor changes for improving user experience.
21st August to 28th August	Submission of my final work to the mentors.

Progress tracking will be done based on the above timeline. The 6 day buffer at the end will be used up for unexpected issues faced during coding period.

After the Mid-term evaluation is done it will be pretty clear if the project will complete in time or will need more work. If it needs more work then I'll have a talk with the mentors to come up with a solution.

I am confident of staying on top of the proposed project timeline. However, if under any circumstance the project is incomplete it will still provide an excellent ground for new contributors to work and learn on as most of the complex stuff will be already done.

September 5
Results announced

Communication:-

So far I've been in touch with the project mentors regularly to understand and learn about the project. If I will be working on this project I plan to update my mentors once every two days giving them complete detail on the things that I've accomplished and also the things I'm currently working on. This will ensure that everything goes according to plan and there is no communication gap between me and the mentors. I also plan to have weekly meetings with the mentors (if they are available) to summarise my work over the week and to learn more about the tech underlying guix.

Qualification:-

My motivation and benefits from this project:

My particular interest was the working of package managers and I have tested quite a lot of them (apt, dnf, zypper and pacman). And then I came across The Guix package manager. I also built a decentralised Voting app in a hackathon (won 2nd prize) using solidity. The concept of decentralisation amazed me and I started looking for projects involving it. And hence my interest in this project. This would be an awesome opportunity for me to learn more about decentralised web and package managers and apply some of that into a real world application.

Plans after GSoC:

As the project revolves around my interests, I plan to continue contributing to the project by adding more transport protocols (planning on working on GNUnet next) and robustify the current GUIX setup after the GSoC period ends. I am also interested in conducting performance

evaluations but I'm still in the exploration phase for this part and scoping it out currently with the help of mentors. I loved the GUIX community support and would love to build with you guys. I plan to be an Open Source contributor forever because I have personally learnt a lot from it and want to give it back to the community. :))

My previous work on Open Source:

Yes, I have worked with oppia-android a year back and I have been an Open Source contributor ever since. I have worked a little bit on gnome-maps as well. I've been consistently contributing in my free time.

Here are a few links from my work in oppia-android:

1. https://github.com/oppia/oppia-android/issues/3591
2. https://github.com/oppia/oppia-android/pull/4280
3. https://github.com/oppia/oppia-android/pull/4283
4. https://github.com/oppia/oppia-android/issues/4288

Skills:

I have a good understanding of networking and decentralisation in general both of which satisfies the skill criteria. I have a good idea of how guix substitutes are packed and I've learnt to use emacs/geiser with guile REPL for optimal development experience. Other than that I have experience working with package managers. I also have good communication skills which is very important for any project. Lastly, I've successfully completed almost all the tasks assigned by my mentor that includes setting up the dev environment and encoding content using guile-eris and guile-cbor into a cbor file.

Skills to learn:

I will have to learn more about the codebase of ERIS and integrate it with SQLite using guile-sqlite bindings. Other than that I'll have to be more familiar with the emacs setup and not to forget The Git Workflow of GNU GUIX.

My Background:

I am a 2nd year undergraduate student pursuing **ICT (Information and Communication Technology)** from **DA-IICT, INDIA**. I have been contributing to open source in my free time for more than a year now. And I've been in touch with the project mentors for more than a month now. They helped me understand more about the GUIX codebase, decentralisation and lots of other stuff related to the project. My past experiences include building android apps, apis using node.js. I've also tried diving deep into the C language and working with hardware.

Very excited to work with you guys!!

EOF