

Computer Science Tripos – Part II – Project Proposal

A strongly consistent index for email using git and MirageOS.

O. Hope, Jesus College

Originator: Dr A. Madhavapeddy

18th October 2018

Project Supervisor: Mr D. Allsopp

Director of Studies: Prof C. Mascolo

Project Overseers: Prof G. Winskel & Dr R. Mortier

Introduction

Maildir is a widely used format for storing emails. Its main benefit is that it uses the filesystem in such a way that client programs do not have to handle locking themselves. The downside of this is that it makes it hard to create a consistent index as we cannot guarantee that the filesystem is in a consistent state when we try to update it. If we did have a consistent index, it would allow for safer concurrent support and the implementation of new features.

MirageOS is a library operating system for building unikernels written in OCaml. This means that it provides OCaml libraries which can be combined and compiled down into high-performance, lightweight kernels. These can then run directly on a hypervisor and only contain the code required to provide the functionality needed. However, the libraries themselves can also be used outside of unikernels to provide functionality to applications on more standard systems. This will be the use case in this project, using the MirageOS libraries to create a standalone application to run on unix-like systems.

The aim therefore is to solve the consistency problem. This can be done by using git, the version control system, to build an overlay on top of maildir in the filesystem. Then multiple filesystem operations can be bundled into commits. These can be used to track of all changes made to the maildir. As these changes are being recorded by a version control system, we can be sure that any index built on top will be strongly consistent. As git also provides branching, we can extend this model to add new features described in the possible extensions section.

We can implement this git overlay using libraries provided by MirageOS which provide git functionality, maildir operations, and even email parsing. With the overlay, and therefore consistent index implemented, we will then be able to make many more guarantees about the state of the maildir at any time. This will also allow for dealing with conflicting operations in an easier and more reliable manner. Furthermore, the overlay will also provide the possibility of easily implementing novel features such as roll-back and separate branches for different use cases.

Starting point

I have used git for version control of software projects in the past, so I have a basic understanding of some of its more commonly used features when using it as a tool. However, I have little knowledge of its more low level actions which will be needed when managing filesystem operations.

MirageOS provides some libraries meaning that I will not have to implement some features from scratch. These include ocaml-git to manage git operations, a library for actioning maildir operations, and one which supports email file parsing.

I have previous experience in Standard ML from the Foundations of Computer Science course in Part IA, but most of the differences and extra features in OCaml will be new to me.

Resources required

For this project I will use my own Apple Macbook Pro (2018) running macOS. I will backup all of my work on a Github repository, and will make weekly backups of my entire computer to an external hard drive. Should my laptop suddenly fail, I have a spare and the option to use the MCS computers in the Computer Laboratory. Alongside my computer, I will take a snapshot of my personal Cambridge email account at the start of the project for testing purposes all other tests will be performed with generated mailboxes. I will require no other special resources.

Work to be done

The project breaks down into the following sub-projects:

1. **Designing overlay functionality:** Working out exactly what git features will be needed for the required overlay functionality, how they fit together, and designing a strategy to implement this.
2. **Prototyping the overlay:** Building a version of the overlay to work using standard command line tools (including git) and non-mirage scripts.
3. **Adding index functionality:** Using the overlay to actually implement the prototype of the consistent indexing.
4. **Adapting the overlay to MirageOS:** Implementing the overlay with MirageOS libraries and OCaml using the knowledge gained whilst prototyping, using pre-existing libraries to manage the git and filesystem operations.
5. **Testing and Evaluation:** Checking that all maildir operations continue to work as expected, testing the index for strong consistency. Testing resolution of conflicts from concurrent operations. Quantifying the slow-down relative to maildir with no consistent index over operations such as insertion and deletion of emails in different quantities and concurrent actions from different processes.

Success criteria

The success of the project will be evaluated against:

1. Implementation of a working git overlay on top of the filesystem in MirageOS.
2. A strongly consistent maildir index using the git overlay which allows all standard maildir operations.

Possible extensions

If I finish both my implementation and evaluation of the project early, then it could be extended by implementing:

- Rollback support in case of user errors such as accidental deletion.
- Branch support to process emails for different platforms. For example a branch for mobile clients which has all images removed from the emails to reduce mobile data usage.
- Intelligently handling conflicts such as replying to the same message twice from different clients.

Timetable

Planned starting date is 22/10/2018.

1. **Michaelmas weeks 2–4 (22nd October–31st October):** Read through the full git documentation of all the standard features, research the maildir format, evaluate different indexing techniques, start learning OCaml, and take a snapshot of my Cambridge email account.

Milestone: Have decided on a high level implementation strategy

2. **Michaelmas weeks 5–6 (1st November–14th November):** Write the prototype overlay using git on a standard OS (macOS) and continue learning OCaml.
3. **Michaelmas weeks 7–8 (15th November–28th November):** Add the indexing functionality, test under all the necessary operations, read the MirageOS documentation relating to the filesystem, git operations, and any other libraries that will be needed.

Milestone: Prototype implementation complete

4. **Michaelmas vacation weeks 1–2 (29th November–12th December):** Start implementation of the overlay in OCaml.
5. **Michaelmas vacation weeks 6–7 (3rd January–16th January):** Continue implementation of overlay in OCaml, and test index is consistent for all operations.

Milestone: OCaml application complete

6. **Lent weeks 1–2 (17th January–30th January):** Write progress report. Start to set up further tests for the application and begin evaluating its features.

7. **Lent weeks 3–4 (31st January–5th February):** Set up and perform tests to evaluate the performance of the overlay under different operations and maildir conditions.

Milestone: Testing of all basic features complete

8. **Lent weeks 5–6 (14th February–27th February):** Implement possible extensions if all previous milestones met and begin writing main dissertation chapters.
9. **Lent weeks 7–8 (28th February–13th March):** Complete dissertation first draft and complete extensions.

Milestone: Dissertation draft complete

10. **Easter vacation (20th March–24th April):** Start revising dissertation using feedback given, and complete any remaining evaluations.
11. **Easter term weeks 1–2 (25th April–8th May):** Complete dissertation final draft.

Milestone: Dissertation complete

12. **Easter term week 3 (9th May–15th May):** Any remaining proofreading, and an early submission to concentrate on exam revision.

Milestone: Dissertation submitted