A Project

entitled

combox

by

Siddharth Ravikumar

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Computer Science

Dr. Robert C. Green II, Committee Chair

Dr. XX, Committee Member

Dr. XX, Committee Member

Dr. Michael Ogawa, Dean
College of Graduate Studies

Bowling Green State University
May 2016

Public Domain, No Rights Reserved.

Siddharth Ravikumar has dedicated the work to the public domain by waiving all of his rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law. You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

See https://creativecommons.org/publicdomain/zero/1.0/legalcode for the full legal verbiage.

An Abstract of

combox

by

Siddharth Ravikumar

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Computer Science

Bowling Green State University
May 2016

File storage providers on the Internet have made it non-trivial for individuals to store personal files on the file storage provider's computers. After Mr. Snowden disclosed information about the National Security Agency' (NSA) surveillance programs that allowed the NSA to access information stored on file storage provider' computers, online file storage became a non-solution for storing personal files for everyone who detested the possibility of somebody else being able to access their personal files. In the past, there have been separate efforts to come with a solution to allow individuals to use storage space provided by file storage providers in a way that it made it impossible for file storage providers and to access the files. combox is one such effort. It allows an individual to store personal files in the "combox directory" on all her computers (running GNU/Linux or OS X) and the combox program takes the files, splits and encrypts them and spreads them across file storage providers' directories. Therefore, when an individual uses storage space provided by file storage providers through combox, each file storage provider gets only a part of the file in an encrypted form.

Dedicated to the `$EDITOR` I use to literally write everything.

Acknowledgments

Dr. Robert C. Green II who gave me an opportunity to work on combox.

Contents

List of Tables

List of Figures

List of Abbreviations

YAML	YAML Ain't Markup Language

Preface

42.

Chapter 1

Introduction

Chapter 2

Background

Chapter 3

Literature Review

Chapter 4

Architecture and Design

Chapter 5

Testing

5.1 Unit testing

The `nose`[1] testing framework was used to write unit tests for the functions and classes part of the `combox.config`, `combox.crypto`, `combox.events`, `combox.file`, `combox.silo` `combox._version` modules. Unit tests were not written for `combox.cbox`, `combox.gui`, `combox.combox.log` modules.

Unit tests for combox become reality by pure serendipity. During the time, when I started working on combox, I was learning to use the `nose` library to unit test python code. Since, `combox` was being written in python, I started making it a norm to write unit tests for functions and classes in combox modules.

As mentioned before, unit tests were not written for some modules either because it would make no sense to write one (for the `combox.cbox` module, for instance, which basically uses functions and classes defined in other modules to run combox) or it was not clear how to write unit tests it (the `combox.gui` contains just the `ComboxConfigDialog` a graphical front-end which uses the configuration function defined in the `combox.config` module to complete the combox configuration based on the user input).

It must be noted here that pure Test Driven Development (TDD) was not observed

– most of the time the function/class was written before the its corresponding test was written.

5.1.1 Benefits

While writing unit tests definitely increased the time to write a particular feature, it enabled me to immediately check if a feature worked as it should for the given use case or given set of inputs.

With the benefit of hindsight, unit tests greatly helped in testing the compatibility of combox on OSX. Before the `v0.1.0` release, combox's node directory monitor always assumed that a file's first shard (`shard0`) is always available; while this assumption did not create any problems on GNU/Linux, on OS X, this assumption made the node directory monitor to behave erraticly – this issue (bug #4[2] was immediately found when the unit tests were run for the first time on OS X. Another instance where unit tests helped was just before the `v0.2.0` release; major changes, including the introduction of file locks in the `ComboxDirMonitor`, were made to the `combox.events`. When the unit tests were run OS X, two tests failed, revealing a difference in behaviour of watchdog[3] on GNU/Linux and OS X on file creation[4]; without unit tests, there is a high probability that this bug would never have been found by now.

5.1.2 Caveats

Unit tests are helpful in testing the correctness of a feature for `N` number of use cases but it does not necessarily mean the written feature correctly behaves for use cases that the author of the feature did not consider or did not think about while writing the respective feature. As Dijkstra correctly observed[5]:

Testing shows the presence, not the absence of bugs

Unit tests failed reveal bugs #4, #5 #6 #7 #5 #10 #11[2]; these bugs were found when manually testing combox.

5.2 Manual testing

5.3 Stress testing

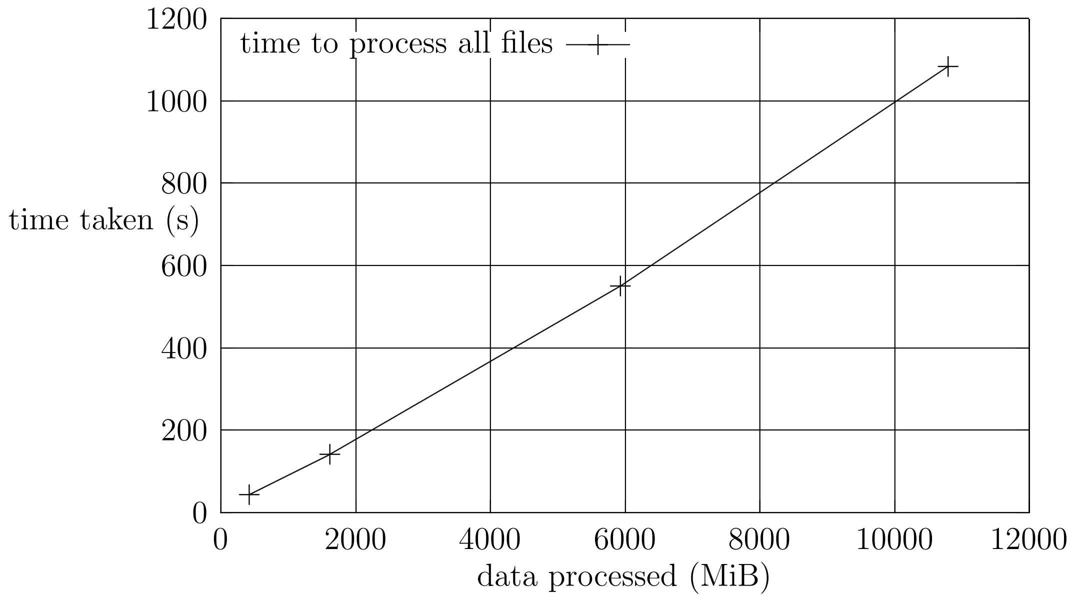

Figure 5-1: time to process all files

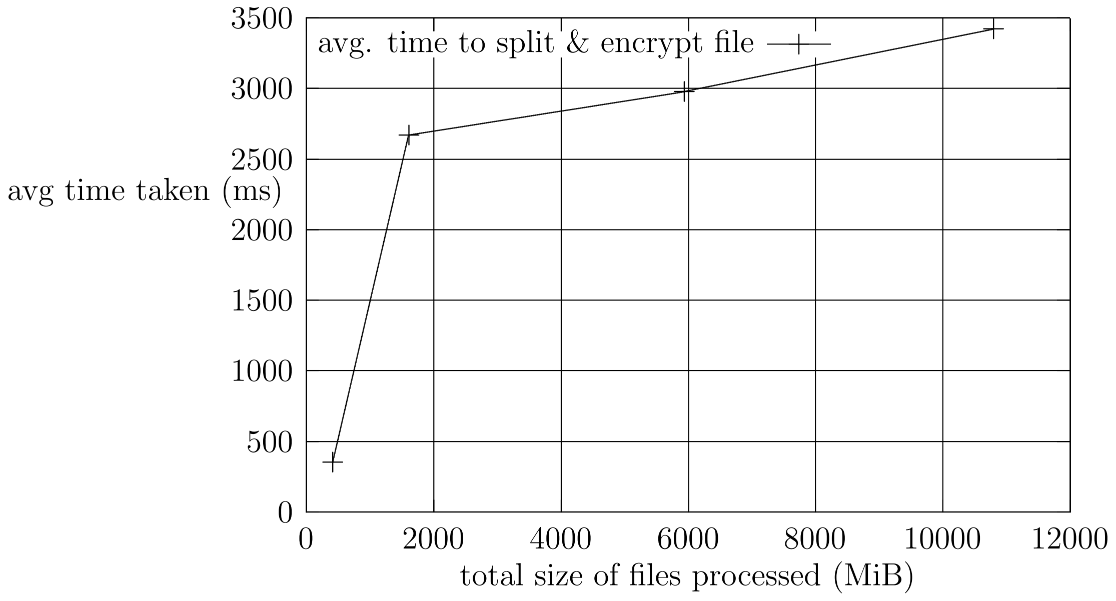

Figure 5-2: avg. time to split and encrypt

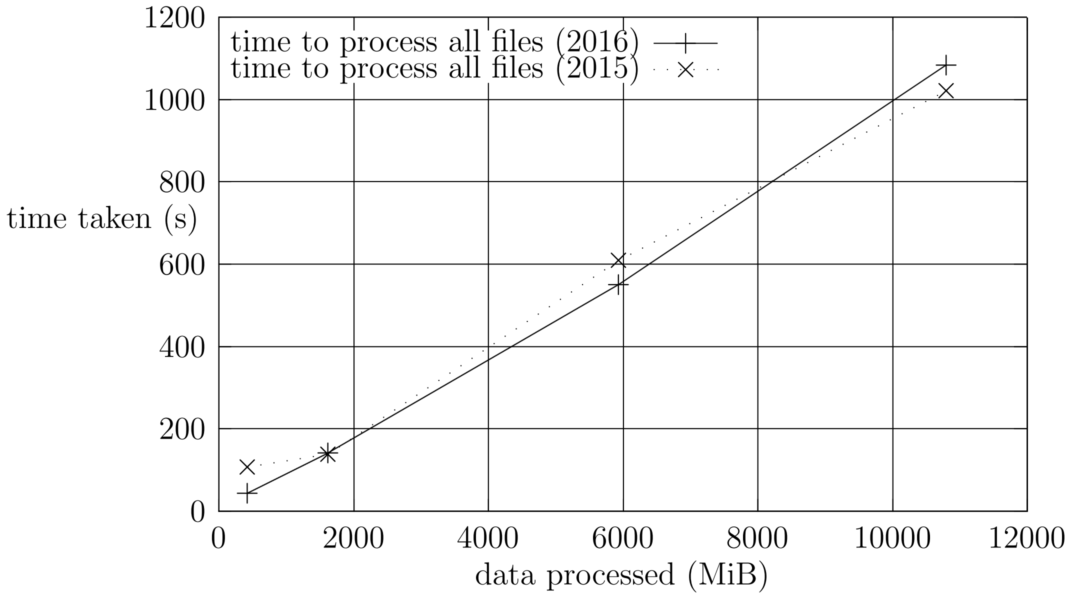

Figure 5-3: time to process all files - difference between 2015 and 2016

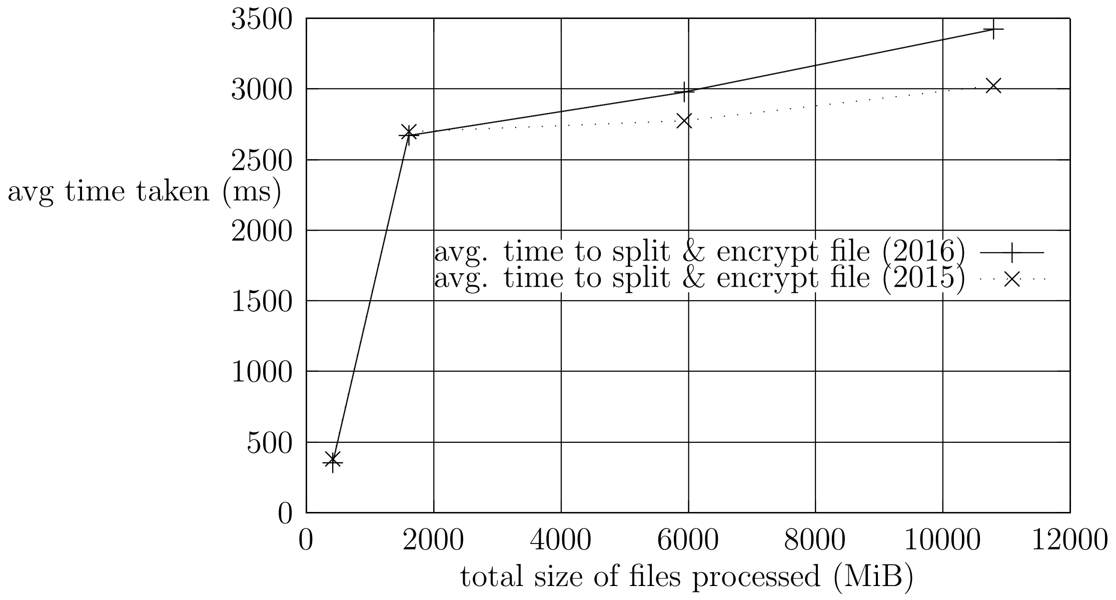

Figure 5-4: avg. time to split and encrypt - difference between 2015 and 2016

Chapter 6

Conclusion and Future Work

References

[1] “Nose - a nicer testing for python.” [Online]. Available: https://nose.readthedocs.org/en/latest/

[2] “combox issue tracker (org-mode).” [Online]. Available: https://git.ricketyspace.net/combox/plain/TODO.org

[3] “Watchdog - python api library and shell utilities to monitor file system events.” [Online]. Available: https://pythonhosted.org/watchdog/

[4] “combox - watchdog ’file create event’ bug fix.” [Online]. Available: https://git.ricketyspace.net/combox/commit/?id=8c86e7c28738c66c0e04ae7886b44dbcdfc6369e

[5] J. Buxton and B. Randell, “Software engineering techniques,” NATO Science Committee, Tech. Rep. p. 16, 1969. [Online]. Available: http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1969.PDF