haute école
neuchâtel berne jura

arc

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

Bachelor Spring Project

HE-ARC 2016

Overclouds

Author:
Romain Claret

Supervisor:
Marc Schaefer

April 26, 2016

Abstract

Overclouds is a project whose goal is to create an anonymous and decentralized internet data sharing service right through the browser.

1 Description

1.1 English

The initiative behind the project is to create a new generation of internet data sharing tools, suited for today's paranoia for privacy on the internet and the preservation of knowledge for the next humanity generations.

The idea is to give the ability to the user to not rely on corporate servers, or farms of servers anymore. On Over Clouds, everybody and everything are now anonymous nodes, and they connect one to another freely and anonymously.

The network is a democratic mesh of nodes. The data is moving from a node to another across the network via other nodes and is ruled by the consensus of users.

We are aiming that users only need to have a standard Internet connection and a browser with JavaScript capabilities to use the service.

1.2 French

Must to the translation when the English part is validated

Contents

2 Preface

2.1 Introduction

TODO

2.2 The Big Picture

TODO

2.3 Objectives

TODO

2.4 Specifications

TODO

2.5 Management

TODO

2.6 State of the Art

TODO

2.6.1 Similar products (Existing Networks)

TODO

2.6.2 Transfer Protocols

TODO

2.6.3 Protection

TODO

2.6.4 Cryptography

TODO

2.6.5 Hardware

TODO

2.6.6 Block-Chains

TODO

2.6.7 Decentralized applications

TODO

2.6.8 Reputation Management

TODO

2.6.9 Operating Systems

TODO

2.6.10 Technologies

TODO

3 Analyses

3.1 Communication

TODO

3.2 Cryptography

TODO

3.2.1 Compare

Table 1: Hashing 0-10MB Files /milliseconds based on [1]

Libraries	Sha1 (size)	Sha1 (hash)	Sha256 (size)	Sha256 (hash)
sjcl	- - - -	- - - -	- - - -	- - - -
crypto-js	- - -	- -	-	- - -
forge	+	+	+ + + +	+ + + +
crypto-browserify	+ +	+ +	+ + +	+ + +
crypto-mx	null	null	+	- -
git-sha1	+ + +	+ + +	null	null
jshashes	-	-	- -	- - -
rusha	+ + + +	+ + + +	null	null

See Figures 1, 2, 3, 4

Table 2: Key Derivation (pbkdf2) based on [1]

Libraries	Sha1 (time)	Sha1 (size)	Sha256 (time)	Sha256 (size)
sjcl	+ + + +	+ + + +	+ + + +	+ + + +
crypto-js	- - - -	- - - -	- - - -	- - - -
forge	+ + + +	+ +	+ + + +	+
crypto-browserify	+ + + +	+ +	+ + +	- -

See Figures 5, 6, 7, 8

blake2s It is a new algorithm designed specifically for performance and is the fastest implementation. **rusha** is close behind it, and forge's *sha256*.

Note that the above implementations display nearly a completely linear performance.

Table 3: Hashing Small Files /milliseconds based on [1]

Libraries	Sha1 (size)	Sha256 (size)
sjcl	- - -	- - -
crypto-js	- -	- -
forge	+ +	+ + +
crypto-browserify	+ + +	+ + + +
crypto-mx	null	+
git-sha1	+	null
jshashes	-	-
rusha	+ + + +	null

See Figures 9, 10

Table 4: Fastest Hashes /milliseconds based on [1]

Libraries	Sha1 (size)	Sha256 (size)	blake2s (size)
forge	+ + + +	null	null
rusha	null	+ + + +	null
blake2s	null	null	+ + + +

See Figure 11

4 Implementations

4.1 Communication

TODO

4.2 Cryptography

TODO

5 Evaluation

5.1 Tests

TODO

5.2 Results

TODO

5.3 Technologies Recommendations

TODO

6 Conclusion

TODO

7 Bibliography

References

[1] Dominic Tarr. Performance of Hashing in Javascript Crypto Libraries., 2014.

8 Annexes

Add other crypto libraries using **Dominic Tarr**'s benchmark set [1].

8.1 JS Cryptography Library Graphs

TODO

The graphs from the following figures have been made by **Dominic Tarr** [1]

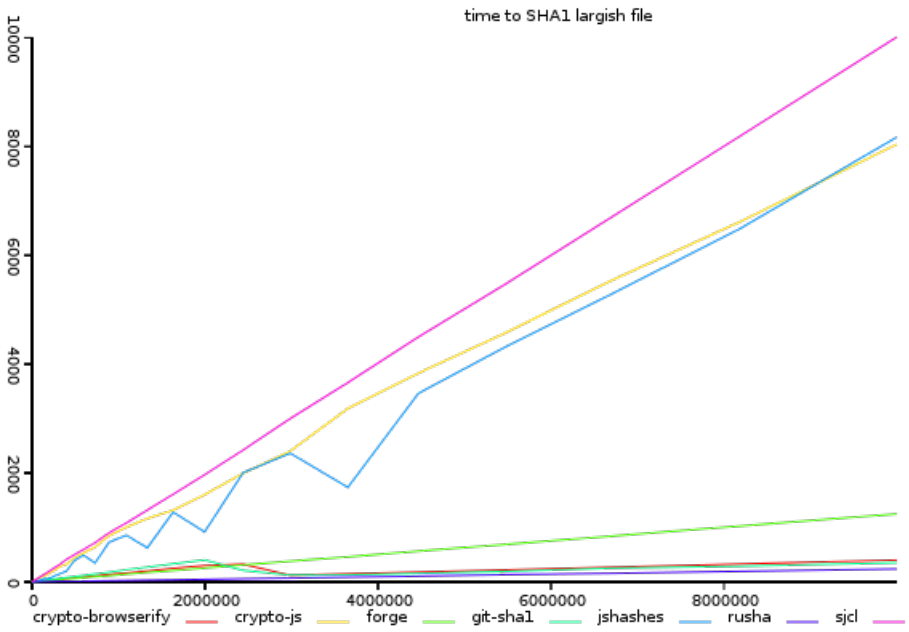

Figure 1: *y-axis shows total time taken, lower is better*

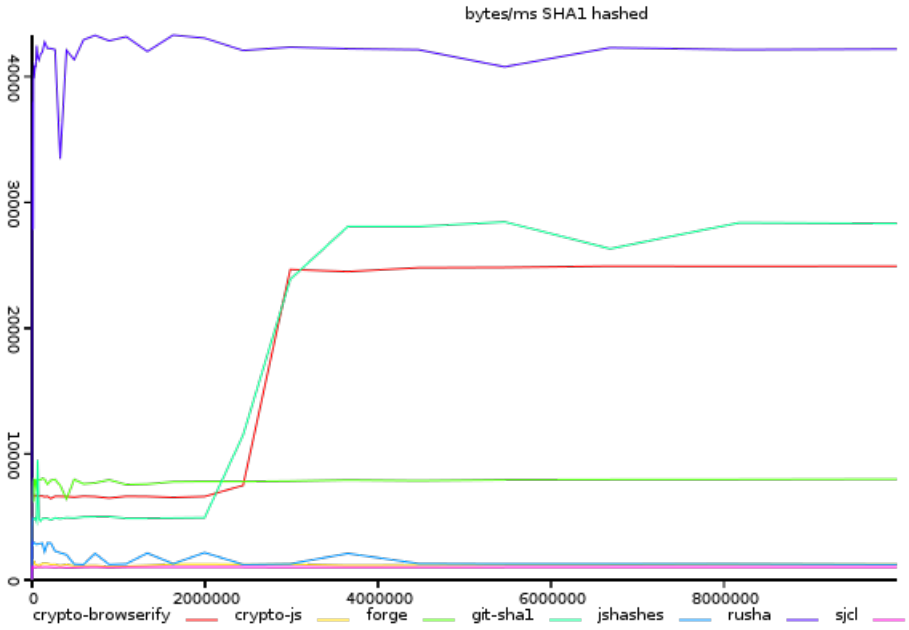

Figure 2: *y-axis shows size/time, higher is better*

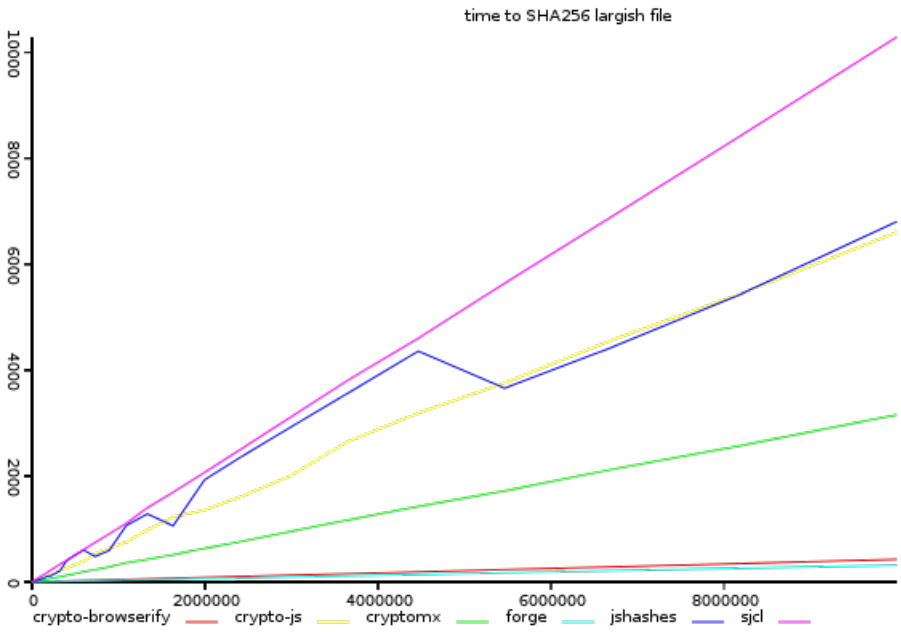

Figure 3: *y-axis shows total time taken, lower is better*

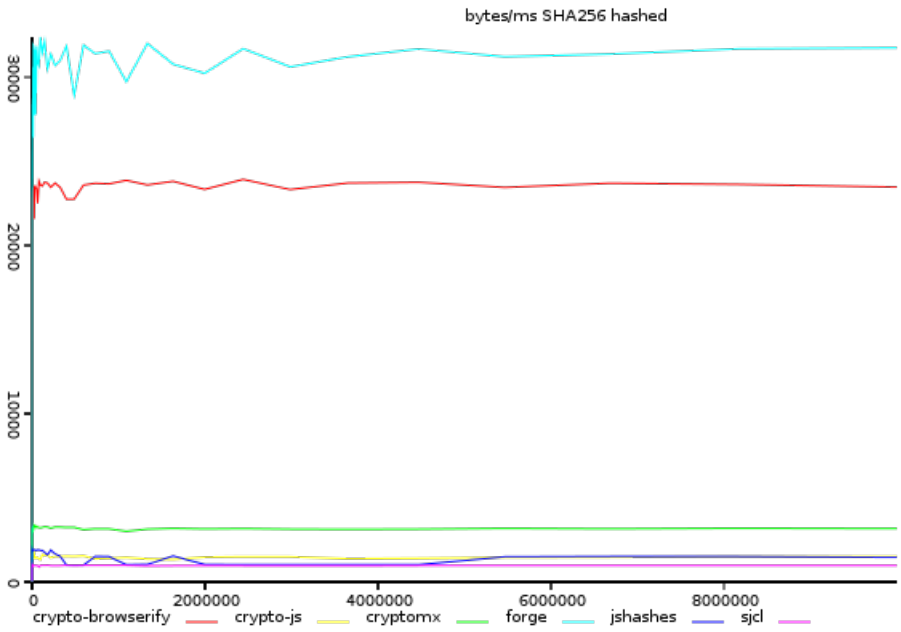

Figure 4: *y-axis shows size/time, higher is better*

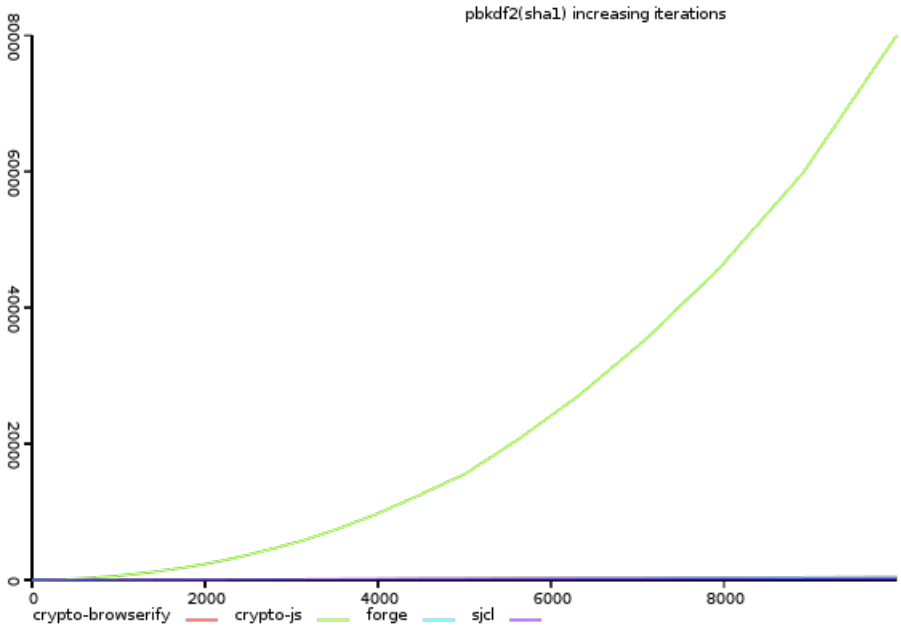

Figure 5: *y-axis shows total time taken, lower is better*

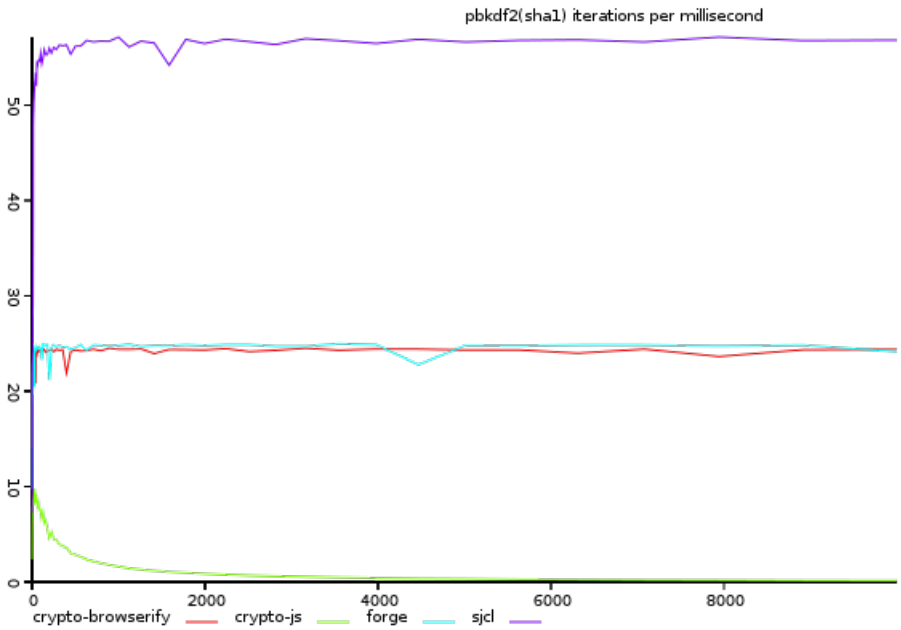

Figure 6: *y-axis shows size/time, higher is better*

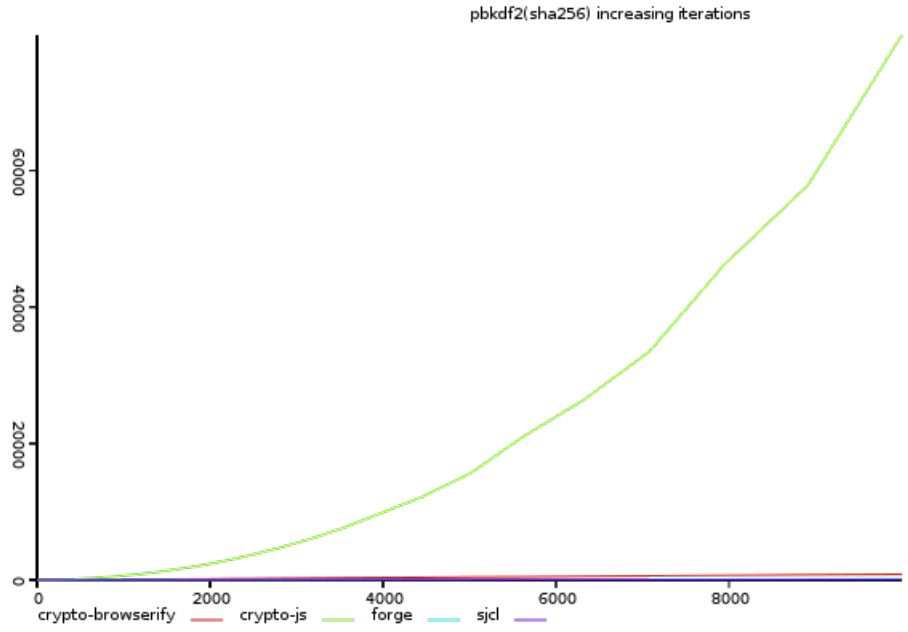

Figure 7: *y-axis shows total time taken, lower is better*

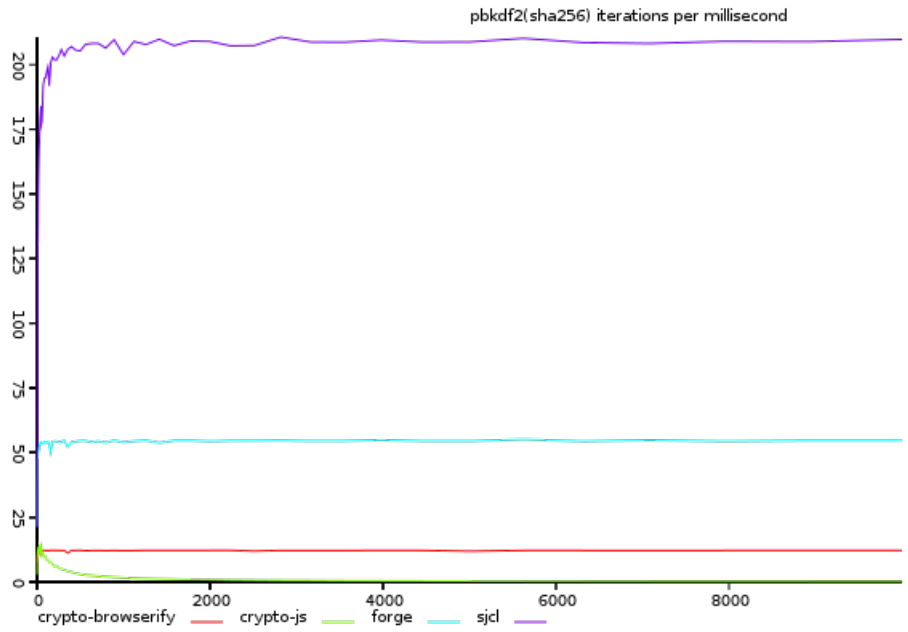

Figure 8: *y-axis shows size/time, higher is better*

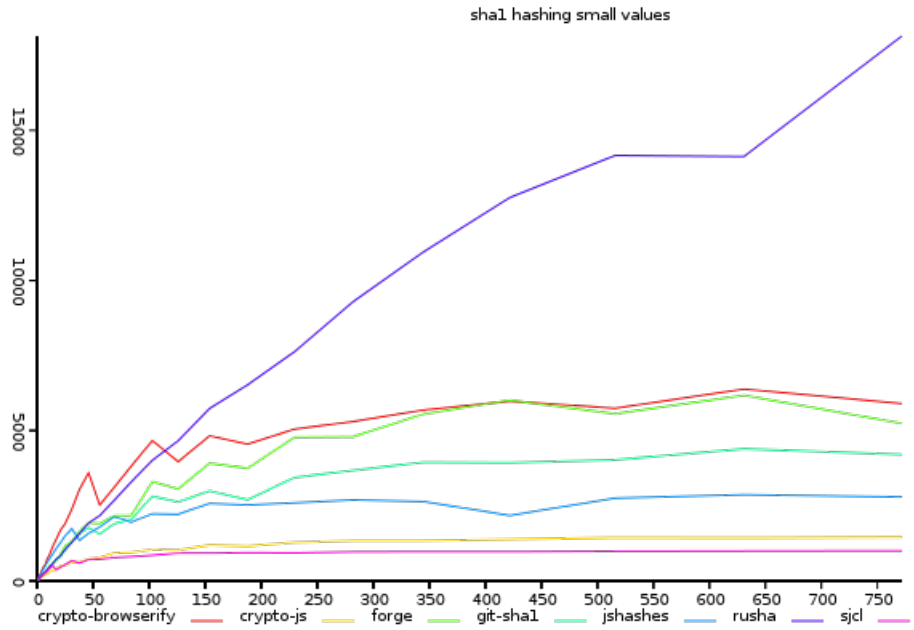

Figure 9: *y-axis shows size/time, higher is better*

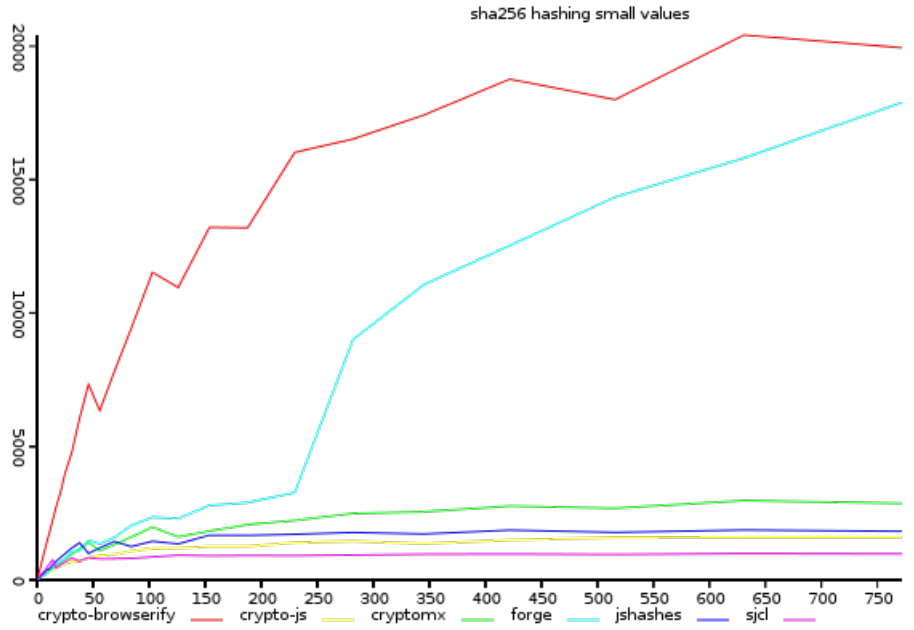

Figure 10: *y-axis shows size/time, higher is better*

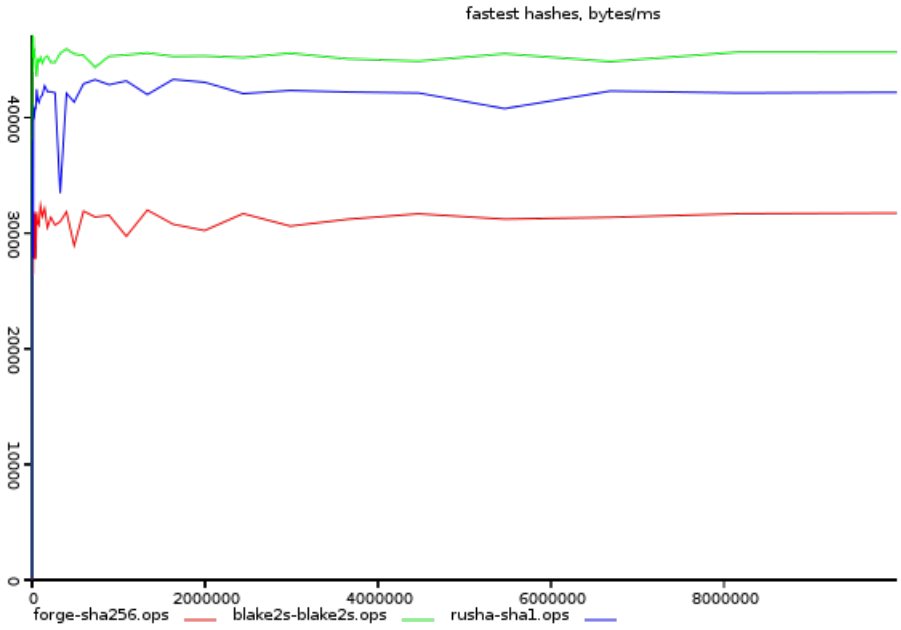

Figure 11: *y-axis size/time, higher is better*