

Sia Load Test Plan

Status: Open for community review (2018-02-07)

Author: Michael Lynch, blogger at [SpaceDuck](#)

Reviewers

- Luke Champine, CTO of [Nebulous](#)

Objective

Determine the maximum renter storage capacity of a single Sia node.

Background

The Sia community does not have empirical data about how much data a single Sia node can rent. Several Sia users have uploaded data to Sia, but nobody has published results of a rigorous experiment to determine Sia's limits.

An accurate estimate of Sia's per-node capabilities will help the community understand the actual costs of storage on Sia, and will help third-party developers build solutions on top of Sia.

Test Environment

Sia will run on a Windows 10 PC. Data files for the test will be stored on a Synology NAS device.

PC

- OS: Windows 10 x64
- CPU: Intel i7-5820K @ 3.3 GHz
- RAM: 32 GB
- Disk: 512 GB SSD

NAS

- Synology DS412+
- 4 TB free space

Network

- Local network: 1 Gbps

- Internet: Verizon FiOS
 - Advertised: 940 Mbps download, 880 Mbps upload
 - [Actual](#): 300-600 Mbps download, 8-175 Mbps upload

Sia

- Build: 1.3.1-windows-amd64 (release build)

Sia Configuration

For each test, the Sia node will start with a fresh wallet. Between tests, I will delete all Sia data folders except for the consensus folder, which I will reuse across tests for convenience. I believe there is a low risk of cross-test contamination from re-using the consensus folder.

I will initialize each wallet with a newly generated seed, funded with 5 KS.

Renter Prices (estimated):

Fees for Creating a Set of Contracts:	98.7 SC
Download 1 TB:	18.89 SC
Store 1 TB for 1 Month:	133.9 SC
Upload 1 TB:	27.6 SC

Sia renter price estimates, as of 2018-01-13.

The wallet amount is based on an upper limit of uploading 10 TB of data for a three month contract:

$$98.7 + (27.6 * 10) + (133.9 * 10 * 3) = 4391.7 \text{ SC}$$

I round this 4391.7 SC up to 5 KS to add a bit of buffer if renter prices change between the test plan and the test execution.

The node will call the `/renter` POST API¹, specifying funds to the full wallet balance amount and period to $(4320 * 3)$ blocks.²

Test Format

A Python script (using [pysia](#)) will perform each of the tests. The script will continue uploading files from the test dataset until uploads stop making progress. “Progress” here is defined by uploading ≥ 100 MB of data in aggregate over the past hour.

¹ [/renter API documentation](#)

² This mirrors [Sia-UI's behavior](#).

The script will not begin uploading a new file until < 5 uploads are in progress. An upload is considered “in progress” if the `/renter/files` API³ returns a value < 100 for the file’s `uploadprogress` property.

If the tests exhaust the free space of the NAS, I will manually delete already uploaded files, generate additional data files, and continue the test script.

If the `siad` process crashes or becomes unresponsive to RPCs, I will manually restart it up to 5 times per testcase. After 5 crashes or hangs, the test is considered complete. I will otherwise not restart the `siad` process.

Note that this test does **not** exercise download functionality. I assume that files uploaded successfully can be downloaded with their integrity preserved.

Test Cases

1. Optimal Case

Data consists of files exactly 125828280 bytes (~120 MiB) in size, filled with random data.

This is optimal size for Sia, as each file will be exactly one full data chunk in Sia:

- $\text{chunkSize} = \text{pieceSize} * (\text{dataPieces} + \text{parityPieces})$
 - $\text{pieceSize} = \text{SectorSize} - \text{TwofishOverhead}$ ([source](#))
 - $\text{pieceSize} = 2^{22}$ ([source](#)) - 28 ([source](#))
 - $\text{pieceSize} = 4194276$
 - $\text{dataPieces} = 10$ ([source](#))
 - $\text{parityPieces} = 20$ ([source](#))
- $\text{chunkSize} = 4194276 * (10 + 20)$
- $\text{chunkSize} = 125828280$ bytes (~120 Mib)

2. Worst Case

Data consists of files 1 byte in size.

³ [/renter/files API documentation](#)

3. Actual Data

Data consists of 4.33 TB of actual DVD/Blu-Ray data (raw ISOs and compressed mp4s). Files range in size from ~100 MB to as large as 48 GB.

Outputs

At the end of the test, I will publish:

- Source code of load test script
 - I will publish this to my personal Github, under the MIT license.
 - The repository will include everything needed for another party to reproduce my results or extend the code, including:
 - Documentation
 - Unit tests
 - A [Travis CI](#) configuration
- Result report
 - I will publish a report on a public website detailing the results of this test.
 - For each testcase, the report will include:
 - Total amount of data uploaded
 - Total cost (in SC and USD equivalent)
 - Notable events that occurred during the test (e.g., crashes, unexpected log messages).

Timeline

Coding begins 2018-02-02. Outputs published by 2018-02-16.