

# FUSE Longevity Tests

Mark Morgan, University of Virginia

[mark-morgan@Virginia.EDU](mailto:mark-morgan@Virginia.EDU)

## Abstract

One of the goals of the XSEDE proposal is to support a global file system using open grid standards like RNS and ByteIO. As part of our feasibility testing, we orchestrated the FUSE Longevity Tests to verify that a FUSE driver front-ending a grid file system comprised of those standards would be able to adequately handle a long-running set of standard file system commands. These tests showed that such a set up is indeed possible and that the availability for this set up can be made reasonably high (at least three-9s of availability).

## Description

In order to test the availability of a long running FUSE driver fronting a grid file system, we devised a test set up where a BASH shell script constantly accessing the file system through a FUSE driver could be monitored for failures. This script was allowed to run for a day and a half and a log of all successful and failed file system operations was kept. This document describes the set up and results of that test.

## Experimental Setup

The FUSE Longevity Tests are designed to measure the reliability and fault tolerance of a global file system implemented using the RNS [1] and ByteIO [2][3] grid standards and a FUSE driver as the client interface. For this test, only two machines were involved. The first of these, Cicero was used to host the file and directory services while the second, Sulla, was used as the client mount point for the FUSE driver. Both machines sit at UVa and are 8 core Intel® Xeon® machines with 24 GB and 48 GB of memory respectively. A GigE switch connects these two machines and both run Ubuntu 8.04.3 for their operating system.

For this test, Genesis II [4] was used as both the server and the client. A grid directory structure was created with 5 subdirectories and 5 files each recursively duplicated in structure to a depth of 4 nested directories.

A BASH shell script was then run over this directory structure (and through the FUSE mount) that in turn ran **sum**<sup>1</sup> on each file, copied a small file onto each file, and finally ran **du** over the entire directory structure. These operations were performed in a tight loop and allowed to run for multiple days with the timestamps of each attempt and the results of each attempt recorded in a log file.

## Results

The test shell script ran from Wednesday, 9 December 2009, 16:31:41 EST until Friday, 11 December 2009 at 8:29:57 EST when the test was concluded (a period of 1 day, 15 hours, 58 minutes, and 16 seconds). During this time period the test ran continually performing 121,162 individual operations, or approximately 1 operation every 1.18 seconds, and during that period experienced only three temporary failures. Each anomaly manifested as a failure to run the associated UNIX binaries (**sum** or **cp**) for a small number of consecutive attempts and in each case the failure eventually corrected itself. The three failures are summarized by the following tables (for each table, the time frames listed constitute a contiguous range of timestamps and the only failures; all timestamps for the test period not included in these tables were successful operations).

---

<sup>1</sup> **Sum** is a binary program available with Linux and other UNIX-like systems that calculates a checksum for a file on disk. This tool is a convenient tool for testing read capabilities on a file system as it reads the entire file and does so quickly.

<b>Timestamp (9 December 2009)</b>	<b>Operation</b>	<b>Result of Operation</b>
20:57:24	Sum	Success
20:57:30	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:31	Sum	Failed
20:57:31	Cp	Failed
20:57:32	Sum	Success
20:57:32	Cp	Failed
20:57:33	Sum	Success
20:57:33	Cp	Failed
20:57:34	Sum	Success

**Table 1: First Failure of Longevity Tests**



Timestamp (11 December 2009)	Operation	Result of Operation
05:42:38	Sum	Success
05:42:42	Cp	Failed
05:42:42	Sum	Failed
05:42:43	Cp	Failed
05:42:43	Sum	Failed
05:42:43	Cp	Failed
05:42:43	Sum	Success
05:42:44	Cp	Failed
05:42:44	Sum	Success
05:42:46	Cp	Failed
05:42:46	Sum	Success

Table 3: Third Failure of Longevity Test

## Summary

The results of this longevity test show that the concept of using a FUSE driver as a front-end client for a standards-based grid file system can provide a reliable and highly available means of supporting a global file system. The three failure periods experienced by the longevity test lasted 10, 15, and 8 seconds respectively and constituted a failure rate of only 0.045%, thus giving us an availability of 99.955% or three-9s of availability.

It is also worth noting that in each failure period, the recovery process looked similar. Each time, as the process started recovering, we saw a very short period where read operations (**sum**) would succeed but write operations would fail (**cp**). Further, for all tests, any failures experienced resulted in much shorter operations (in other words, the failed operations tended to complete quickly).

## Works Cited

1. Pereira, M., Tatebe, O., Luan, L., Anderson, T. "Resource Namespace Service Specification." GFD-R-P.101. 1 March 2007.  
<http://www.ogf.org/documents/GFD.101.pdf>.
2. Morgan, M. ed. "ByteIO Specification 1.0." GFD-R-P.087. 31 October 2007.  
<http://www.ogf.org/documents/GFD.87.pdf>.
3. Morgan, M. ed. "ByteIO OGSA WSRF Basic Profile Rendering 1.0." GFD-R-P.088. 31 October 2007. <http://www.ogf.org/documents/GFD.88.pdf>.
4. <http://vcgr.cs.virginia.edu>.