HPSS System Testing

HPSS Users Forum – October 2019
Disclaimer

Forward looking information including schedules and future software reflect current planning that may change and should not be taken as commitments by IBM or the other members of the HPSS collaboration.
Agenda

- Summary
- Continuous Improvement
- Resources
- Automated Process
- GHI Test Cycle
- Releases Available
- What’s Next
Summary

• Full aggregate recall tests were added to the System Test Framework

• The System Test Team is providing value to non-Red Hat users by adding CentOS, Scientific Linux, openSUSE, and Ubuntu clients as HPSS clients to the automated framework

• The automated test effort has been consolidated to allow the team to provide the same coverage with less people

• All System Testers are now testing GHI which allows the team to find more issues than the previous, smaller GHI test team
Continuous Improvement: Introduction

The HPSS System Test team continues to iterate and improve upon its toolset by adding tests for the Full Aggregate Recall feature, performing automated tests on CentOS, Scientific Linux, Ubuntu, and openSUSE, beginning to incorporate the GHI System Test Framework tests into the automated framework, and performing static analysis on HPSS, HPSSFS-FUSE, and GHI using Clang.

<table>
<thead>
<tr>
<th>Component</th>
<th>2018: Number of Automated Tests</th>
<th>2019: Number of Automated Tests</th>
<th>Changes and Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPSS</td>
<td>56</td>
<td>58</td>
<td>1) Clang 2) FAR tests</td>
</tr>
<tr>
<td>HPSSFS-FUSE</td>
<td>79</td>
<td>79</td>
<td>1) Clang 2) CentOS, Scientific Linux, Ubuntu, and openSUSE clients</td>
</tr>
<tr>
<td>GHI*</td>
<td>0</td>
<td>26</td>
<td>1) Clang 2) Incorporation into automated process 90% complete</td>
</tr>
</tbody>
</table>

* Denotes a new item
Continuous Improvement: Test Automation

• 2016
  • System Test Framework (STF) in use

• 2018
  • Increased the amount of test automation since 2016 through the System Test Framework (STF) which uses Jenkins as the process backbone
  • Improved time to value by reducing resources required to test through virtualization and automation
  • HPSS and GHI workloads

• 2019
  • Test automation has reduced the number of test resources by half
  • Frequency of releases has increased through the update process
  • Continuing to add value to the System Test Framework (STF) by adding tests for new HPSS functionality like Full Aggregate Recall (FAR)
  • Added HPSSFS-FUSE testing to our automated build process
  • Increasing the amount of test automation through the GHI System Test Framework (GSTF)
Continuous Improvement: GHI System Test Framework

- The GHI System Test Framework (GSTF) is a suite of tests
  - Tests are run manually
- Coded in Python to improve portability
- Currently covers backup, restore, DMAPI attribute testing, error testing, and file pin testing
- Upgrades and new features are continually being added to the GSTF to help the GSTF find more issues in GHI
Continuous Improvement: Static Analysis using Clang

• For the purposes of this discussion, “static analysis” refers to a collection of algorithms and techniques used to analyze source code in order to automatically find bugs.

• Static analysis is performed on HPSS, GHI, and HPSSFS-FUSE builds using the Clang Static Analyzer tool and is part of our automated process.

• Static analysis has already prevented potential issues in the field including memory leaks, duplicated “free” logic, and unused variables.
Resources: Hardware

• 2018
  • Tested across eight tape libraries from IBM, Quantum, Oracle, and Spectra Logic, including current beta drives and media
  • Tested across a variety of disk arrays including HDD and FlashSystem
  • Tested on IBM Power 8 ppc64/ppc64le servers and System x servers

• 2019
  • Added IBM Power 9 ppc64le servers to suite of supported HPSS servers
Resources: Virtualization

• Virtualization means new test, development, and support environments are configured and setup in less time

• Setting up multiple servers, installing operating systems, configuring disk cache, and assigning resources went from a days-long process to an hours-long process

• Using an average HPSS configuration of three servers (core, mover, client), virtualization allows for three separate instances of HPSS on a single server
Automated Process: HPSS

- Developers check in new code
- The Jenkins process server
  - Performs an automated build with the newly checked-in code
  - Executes HPSS Test Plan (HTP) smoke tests on the build using four systems
    - HTP smoke tests can catch high-level functional errors
  - Executes System Test Framework (STF) tests on the build using three systems
  - The success of this nightly build process is easily checked by viewing the Bugzilla homepage, and verifying the items in the build box are colored blue
- System Test picks up the build
  - Performs ad hoc tests, bug tests, CR reviews and targeted testing, workload testing, etc.
  - When bugs are found, System Test coordinates with Development to recreate the bug and test the fix as it is developed
- When critical issues are found, they are made available in an Update Release
  - The Jenkins server executes HTP tests on the build using three systems
Automated Process: HPSSFS-FUSE

• Developers check in new code

• The Jenkins process server
  • Performs an automated build with this newly checked-in code
  • Executes HPSSFS-FUSE tests on the build using internally-developed tools on four systems running CentOS, Scientific Linux, Ubuntu, and openSUSE
    • Fifty-eight scenario 2 tests, plus additional tests specific to san3p
    • Thirteen NFS4 tests
    • Eight miscellaneous tests

• System Test picks up the build
  • Performs ad hoc tests, bug tests, CR reviews and targeted testing, workload testing, etc.
  • When bugs are found, System Test coordinates with Development to recreate the bug and test the fix as it is developed
Automated Process: HPSS – Faster, Smaller Release Testing

- Ad hoc testing
- Bug tests
- Long-running tests (24+ hours)

HPSSFS-FUSE targeting tests

STF tests run on multiple systems per interface

Fixes quickly get to the System Test Team. When bugs are found during this cycle, a tester coordinates with development so that:
- the fix is system tested as it is developed
- no additional schedule time is required to test

Nightly Build

Developers check in fixes

Release is GA

http://www.hpss-collaboration.org
HPSS Internal Use Only
GHI Test Cycle

- Developers check in new code
- System Test picks up the build
  - Executes GSTF tests
  - Performs ad hoc tests, bug tests, CR reviews and targeted testing, workload testing, etc.
  - When bugs are found, System Test coordinates with Development to recreate the bug and test the fix as it is developed
- When critical issues are found, they are made available in an Update Release
GHI Test Cycle – Faster, Smaller Release Testing

System Testers manually execute GSTF tests

Fixes quickly get to the System Test Team. When bugs are found during this cycle, a tester coordinates with development so that,
- the fix is system tested as it is developed
- no additional schedule time is required to test

Developers check in fixes

- Ad hoc testing
- Bug tests
- Long-running tests (24+ hours)

Update Release

GHI

Release is GA

Update releases are provided as necessary
Releases Available: Test Automation Brings New Code to the Field Faster

• Since the start of 2019, Update Releases have provided Jag6/TS1160/E09 support, data integrity fixes, and fixes important to customers to existing HPSS deployments – critical and new functionality fixes continue to be available more frequently when compared to the old patch release schedule.

• Risk to the release schedule is minimized due to the fact that hardware and software prerequisites between Update Releases for a given version of HPSS do not change. System Test configurations remain static, allowing testing to proceed unimpeded by extraneous setup and configuration activities.

• GHI release 3.0.1 is generally available.

• HPSS release 8.1 is generally available.
Releases Available: HPSS Update Releases in 2019

<table>
<thead>
<tr>
<th></th>
<th>7.4.3p2</th>
<th>7.5.1p2</th>
<th>7.5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Releases</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Average Fixes per Release</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>
## Releases Available: GHI Update

### Releases in 2019

<table>
<thead>
<tr>
<th></th>
<th>3.0.0p1</th>
<th>3.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Releases</strong></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td><strong>Average Fixes per Release</strong></td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Releases Available: HPSSFS-FUSE Update Releases in 2019

<table>
<thead>
<tr>
<th>2.0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Releases</td>
</tr>
<tr>
<td>Average Fixes per Release</td>
</tr>
</tbody>
</table>
What’s Next: Treefrog Testing Overview

• System Test will test the Treefrog project software in the following areas:
  • Installation
    • Ensure the server, client, and copier installation packages can be installed on an unmodified system on all in-scope, supported architectures and platforms
  • Component
    • System Tests are being written against each requirement
    • Tests are reviewed by Houston HPSS System Engineers
  • Endpoint
    • HPSS
    • LTFS-LE
    • POSIX
    • MinIO
  • Workload
  • Documentation review
What’s Next: Treefrog Client Testing

- System Test will test the Treefrog client on:
  - Red Hat Enterprise Linux
  - Windows 10
  - MacOS
    - *out-of-band testing by developers*
What’s Next: Treefrog GUI Testing

• What is Jubula?
  • Allows GUI testing to be automated
    • Faster time to deployment
  • Provided as an add-on to Eclipse, which is used by IBM, Red Hat, and Oracle
  • Tests are written by System Testers from the perspective of the end-user
  • Supported on Windows Linux, and MacOS
  • Command-line clients allow for incorporation into the automated test framework
  • Using OSS test tools means faster time to deployment
What’s Next: Treefrog GUI Testing

• What is Selenium?
  • The Treefrog application software admin gui is browser-based
  • Selenium allows browser testing to be automated
    • Faster time to deployment
  • Used by Typeform, MIT, and Intuit
  • Tests are automated across platforms
  • Supported on Windows, Linux, and MacOS
  • Jenkins support for Selenium
Appendix A: Terminology

• Update Release – quickly getting critical fixes to the field
• Nightly Build – code is built when new code is checked in
• HTP – HPSS Test Plan
• GSTF – GHI System Test Framework
• STF – System Test Framework
• Jenkins – continuous delivery environment, includes automation
• Smoke Test – preliminary tests for catching high-level functional errors
• Bug Test
  • A bug fix is tested by a member of the HPSS System Test Team, who performs focused testing on the change and surrounding functionality inherent to the bug
Appendix B: Nightly Build

• HPSS, GHI, and HPSSFS-FUSE are built nightly when developers have checked in code that same day. HPSSFSFUSE is built in conjunction with HPSS.

• Each nightly build is performed across each supported architecture (x86_64, ppc64, ppc64le)

• The nightly builds are tested by the HTP on two systems and by the STF on three systems
Appendix C: HTP

- The HPSS Test Plan (HTP) is an existing suite of baseline, functional, and workload tests.
- The HTP is owned by the HPSS Support Team.
- The System Test Team uses seven HTP test scenarios for smoke testing HPSS nightly builds and update releases.
  - Smoke tests – preliminary tests for catching high-level functional errors
- The HTP provides the System Test Team with a suite of robust tests it can use without having to code these tests again.
Appendix D: System Test Framework

• The System Test Framework (STF) is a suite of tests
  • It’s fully automated!
  • Tests run using Jenkins

• Coded in Python to improve portability

• Currently covers regression testing, functional testing, and error scenario testing.

• Upgrades and new features are continually being added to the STF.

• The STF utilizes fifty-eight test scenarios when testing the HPSS nightly builds.
Appendix E: Release Nomenclature

• Version notation
  • Major.Minor
  • Major.Minor[u]Update
  • Major.Minor[e]Efix

• Major Release
  • Metadata and API changes

• Minor Release
  • No metadata or API changes

• Update Release
  • Can be used to address issues found in the field
  • Can contain critical enhancements
  • Can contain bug fixes

• Efix
  • Used to address major issues
The End