

Whats New

Release date - June 30, 2022

Main Highlights

1. **Time-period reporting adds a new dimension to the performance metrics.** Now you can generate a performance report for any test execution period. For example, if a test consists of a VU ramp-up phase, steady load phase, and ramp-down phase, you can analyze performance metrics for each of these phases separately and combined. 🔄
2. **Test result recovery.** Computer crashes happen, and unsaved data can be lost. But now, if your system crashes during test execution, all test data is preserved. Simply open the test result after re-starting StresStimulus, and it will calculate test results using data captured up to the minute of the crash. 🔄
3. **Data size limit removed.** In previous versions, StresStimulus used the embedded SQL CE database repository with a maximum of 4GB of storage per test run. It was sufficient to run most of the tests for 4-8 hours with full logging or longer with partial logging. This limit is removed in v 5.6. StresStimulus now comes with SQLite, a more robust data engine that does not have a data size limit. 🔄

Broader support for recording from any source

1. **Automated recording of non-browser application scenarios.** StresStimulus can start a desktop application with necessary command-line parameters and record traffic from the moment of launching a specified process. 🔄
2. **Better support for recording non-browser Java applications.** The necessary proxy settings for this case are applied automatically. 🔄
3. **Better support for recording applications listening on the localhost.** The option to capture requests to localhost is available now. 🔄
4. **More informed scenario re-recording.** You can keep your notes directly in the recorder to make subsequent recordings easier. Your notes will be available next time you record the same scenario or work with the same test. 🔄
5. **Added support for recording Server-Side Events web applications.** 🔄

Easier to use UI

1. **More intuitive mix-weight control.** A new tree control simplifies VU distribution across load agents and test cases/test case groups. 🔄
2. **Support for more meaningful test case names.** Now test case names can have up to 55 characters. The previous limit was 32.
3. **A more consistent layout of the workflow tree.** The test profile-related settings are now displayed under the Test Profile node.
4. **Speedup frequent operation of changing the test profile just before launching a test.** Pre-run test dialog now has a test profile selector. 🔄
5. **More informative UI.** For example, the selected test profile name is displayed in the Workflow tree. 🔄

More elaborate scripting

1. **Faster troubleshooting extractor misconfigurations.** The extractor verification tree doesn't only display extractor run-time execution data but also its design-time definition, simplifying troubleshooting. 🔄
2. **Easier extractor management.** If the extractor verification tree shows a playback error caused by an extractor, the user can delete it in-place right from this tree. Previously, users needed to navigate to the extractor tree and find the faulty extractor, which took more time. 🔄
3. **More robust JPath extractors.** Now a JPath extractor can return objects along with primitives. 🔄
4. **Added an option to group test cases into several tiers to emulate complex timing and concurrency rules.** Every test case or test case group can be assigned to a tier that starts at an appropriate moment representing a certain test phase. For example, the unexpected peak operation phase follows a regular operation phase. 🔄
5. **Broader parameterization coverage.** Parameterization datasets now support newline characters, so users can parameterize multi-line data. 🔄
6. **The extractor designer can test dynamic extractors.** Previously dynamic extractor could be tested only during the test case Verify operation.

More robust test execution and automation engine

1. **Test duration limit is removed.** Now a test can run for up to 500 hours which is sufficient for any test. 🔄
2. **More robust execution error handling.** A new configuration option allows, in the event of an error, to stop the current iteration, complete the logout transaction, and start a new iteration. 🔄
3. **Simplified and more reliable distributed testing with a centralized database repository.** In the previous versions, every load agent must have a speed-sensitive connection to the SQL Server. In this version, such connections are no longer required, which eliminates database connection failures. 🔄

4. **Ability to gracefully stop tests initiated manually from a command line or CI/CD pipeline.** For tests that started in *noui* mode, press Ctrl+C or Ctrl+Break to initiate the test completion and calculate the result. 🔄
5. **Better cross-machine collaboration in distributed testing.** Load agents will automatically stop if the controller suddenly loses connection to them due to crashes or network issues. 🔄
6. **Support for a single VU to execute multiple server credentials.** TC Group can use the child test case's user group enabling a VU to execute multiple test cases, all with different server credentials 🔄

Cloud testing got easier

1. **Support for all modern instance types available in AWS.** AWS auto-provision wizard now provides a broader selection of cloud instance types.
2. **Simplified selection of cloud instances.** Auto-provision wizard now allows filtering of the extensive master-list of available AWS and Azure instances to simplify selection. 🔄🔄🔄
3. **Added monitoring of cloud load agents.** Now you can monitor system resources of load agents created automatically before the test starts. Previously, you could monitor system resources only of the load agents created manually. 🔄
4. **More choices of instantiating cloud load agent.** Now you can decide to auto-destroy load agents at the test-end and auto-provision on-demand before the next execution or reuse them in subsequent tests. 🔄🔄🔄

Extended performance analytics

1. **Added real-time transaction time percentiles.** Now run-time dashboard displays the 90%, 95%, and 99% transaction time percentiles during the test execution. Previously, the percentiles were available only on the final report. 🔄
2. **Added a threshold for monitoring sub-optimal transaction performance.** You can now specify the expected time for the transaction response (required transaction time). Previously StresStimulus registered only below-acceptable transaction time (goal). 🔄🔄🔄
3. **Added remote load agent error information to the run-time dashboard.** Previously the run-time dashboard showed only local errors. 🔄
4. **More transaction details on the report.** Transaction and Page detail grids now display the number of started and failed iterations and the number of completed iterations.
5. **Reusable performance monitoring settings for saving configuration time.** Performance monitoring configuration can now be exported from one test to another. 🔄
6. **Doubled the maximum number of monitoring graph panels to 16.** 🔄
7. **More flexibility in generating performance metrics.** Previously, StresStimulus automatically calculated test results upon test completion. Now users can recalculate test results on demand. 🔄
8. **More customized graph presentation.** Now you can select a subset of the most crucial performance curves to be visible at all times. On the subsequent report reopening, curves visibility will persist. 🔄
9. **Faster test result opening.** Memory management is optimized by loading only the data points in visible curves. Other curves are loaded on demand when the user unhides them. 🔄

Miscellaneous Improvements

1. Dynatrace integration is upgraded to support the latest header specification 🔄
2. Obfuscated machine passwords and connection strings in the configuration file for increased security. 🔄 🔄 🔄
3. Multiple fixes and improvements.