TechBulletin - Enabling SCoE Affects IE Agent Response Times

Information:

Environment

In order to capture, store and, on error, forward screen capture data, the IE Browser Agent must perform certain actions that incur incremental overhead vs. a non-SCoE enabled test. This overhead is caused by the I/O costs attributable to reading the required data objects from the IE Browser (read operation on an internet file descriptor) and writing that data to file. This creates the "screen capture" that will be packaged and stored in the event of an error.

Symptoms

While this read/write overhead is incurred only when SCoE is enabled, it is incurred regardless as to whether or not there is an error.

Solution

The original IE Agent design performs the read/write operations within the scope of the transaction response measurement timers. After a careful analysis, engineering has determined that the current design does not accommodate moving these operations outside the scope of the timer due to the "black box" nature of IE and where we are able to insert our "hooks" into the WinInet.dll to drive the browser, capture the timing information and record the screen data. Multiple approaches have been devised and prototyped to reduce/eliminate the overhead in an attempt to ensure SCoE enabled tests will have the same response time results as SCoE disabled tests.

At this juncture, the only viable improvement that can be made without a lengthy, major re-design of the IE Browser Agent is to minimize (not eliminate) the overhead by writing the data to an in-memory buffer (minimize the write overhead), then write the contents of the buffer to file outside of the timer scope. The data read operation must still be performed in the scope of the transaction due to the nature of the WinInet.dll library capabilities available. The data read operation does still impose some incremental overhead that varies depending on the type & quantity of data within a given operation/page.

Engineering has implemented the improvement and has performed unit and functional testing to confirm that improvements are realized. Initial testing demonstrates that the response time overhead has been reduced to ~5%, on average, for the test cases and test nodes utilized in this testing.

Root Cause

Further characterization and analysis is required to fully understand the overhead by object type/size, node geography, etc.

⚠