THINMANAGER
A Rockwell Automation Technology

Stratus ztC Edge Characterization

Overview

The Stratus ztC Edge is a reliable, rugged, compact computing solution designed for running virtualized industrial applications. Comprising both hardware and software, it has been designed to live and operate in the manufacturing environment as a standalone virtualization computing engine. A redundant pair of units share a proprietary hypervisor OS that allows utilization of resources from either unit (load-balancing), while still providing hardware redundancy in the event one unit fails. See the datasheet at the following link for more system specifications – https://www.stratus.com/assets/SB-ztC-Edge-OEM.pdf. The ztC Edge is well suited for OEMs because it enables the rapid and easy delivery of reliable, industrial computing functionality, increasing efficiency, reducing cost, and lowering downtime risk.

Stratus ztC Edge and ThinManager

The ztC Edge is designed as an Operations Technology (OT) centric virtualization solution for the plant floor. Management and maintenance of the ztC Edge system is self-contained and designed to be very straight-forward – putting the system in the hands of the OT personnel. ThinManager software is a visualization and mobility solution designed to centralize device and application management and securely deliver that content to a wide range of computing end devices. The ThinManager solution is the perfect one to pair with the Stratus ztC Edge because it too is designed as an OT centric solution – allowing the management and maintenance of the system to be in the hands of the OT personnel. Together ztC Edge and ThinManager can revolutionize how OEMs/end users design, develop, deliver, operate, and support their machine visualization applications.

Stratus ztC Edge and ThinManager Application Characterization Guidelines

The goal of this document is to describe how well the ztC Edge performs when hosting FactoryTalk View SE HMI application(s) being delivered to thin clients with ThinManager. For this characterization, we have set up three virtual machines to be run on a ztC Edge 100i system: 1 domain controller (DC), 1 FactoryTalk View SE HMI server (HMI), and 1 ThinManager/RDS server/FactoryTalk View SE HMI Client (RDS). The ztC Edge 100i has available 8 vCPU cores and 32GB of virtual memory to allocate to virtual machines. Following the instruction of the Stratus team, the DC virtual machine was allocated 1 vCPU and 2 GB vRAM. The HMI virtual machine was allocated 2 vCPUs and 4 GB vRAM. Finally, the RDS virtual machine was allocated 2 vCPUs and 8 GB vRAM. We believe this to be a realistic usage of the ztC Edge's computing resources.

Stratus ztC Edge and ThinManager Application Characterization – Phase 1

Phase 1 of testing was completed running both the InstantFizz and the CookieFactory demo HMI applications on the HMI server. The logic for these applications was also executed on the HMI server using Studio 5000 Logix Emulate. 16 RDS sessions were launched and delivered to thin clients, 8 from the CookieFactory HMI application, 7 from the InstantFizz HMI application, and 1 Studio 5000 Logix Designer application. The plan of attack for testing was very straightforward – keep launching RDS sessions until latency is apparent.

The ztC Edge handled the load very well. Noticeable delays when navigating overview screens began to occur with variable frequency with 15 or more sessions running. The more resource intensive features of FactoryTalk View SE like alarming and trending with datalogging showed as-expected non-delayed responses up until that point and the high-speed trending within Studio 5000 Logix Designer performed well. We ran the system in a load-balanced configuration as would be optimally utilized by OEMs and end users for a deployed solution. The DC and HMI images were run on node 0 and the RDS image was run on node 1 of the ztC Edge system.

The results of phase 1 tests are very promising as this characterization was designed to mirror how the ztC Edge system with ThinManager would most likely and optimally be deployed by OEMs for production. The use of load balancing would be best practices for runtime performance, and the use of lighter-weight HMI applications would be typical of an OEM. To clarify, an OEM FactoryTalk View SE application should be considered lighter-weight when compared to a system running full-scale process control like a paper machine. The tags on scan, alarming, trending, data-logging, eventing, volume of displays, number of clients, etc. are expected to be less for an OEM solution versus a full process system solution. Specifically the CookieFactory application has 46 displays with 529 total unique tags and 10 total alarms. The InstantFizz application has 53 displays with 272 total unique tags and 5 total alarms. Reaching 15 sessions before performance degradation is significant, because we anticipate this to surpass an expected session count that a typical OEM would deploy with a particular machine's visualization applications. Please continue reading for Phase 2 testing description and results. This section will describe the metrics used to conclude how overall performance of the system was evaluated. Phase 1 statistics graphics are included on pages 5 and 6. The comparison of Phase 1 to Phase 2 graphics prove an effective method to visualize the ztC Edge's performance, and differences between the two tests.

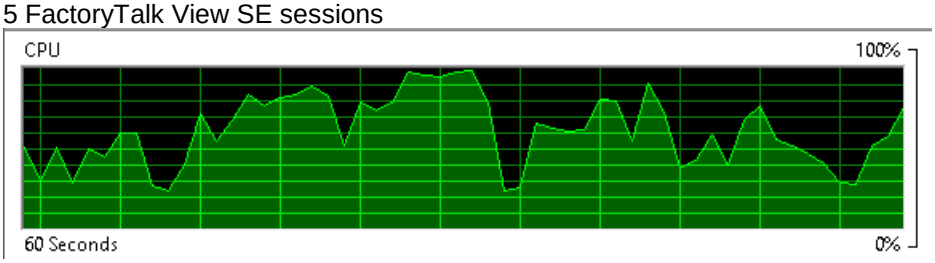
Stratus ztC Edge and ThinManager Application Characterization – Phase 2

Phase 2 of testing involved greater stressing of the ztC Edge system when compared to phase 1. A significantly larger HMI application containing 413 displays with 25166 total unique tags and 1025 total alarms was run. Additionally, to simulate the failure of one of the units in the pair, all three virtual machines were run on node 0. The logic for the application was run on a ControlLogix L75 controller. As previously stated, an average OEM HMI application would not be this large. The application used contains the PlantPAX process objects library which extensively utilizes global objects and overview screens containing hundreds of tags on scan at once. The application contains 8 process cells, each of which has an overview screen containing 1172 tags. By contrast, the overview screen in the CookieFactory application contains 99 tags. Phase 2 of characterization was designed to stress the ztC Edge beyond its anticipated use case, specifically to find the number of RDS sessions which caused noticeable delay when navigating screens and viewing live tag updates.

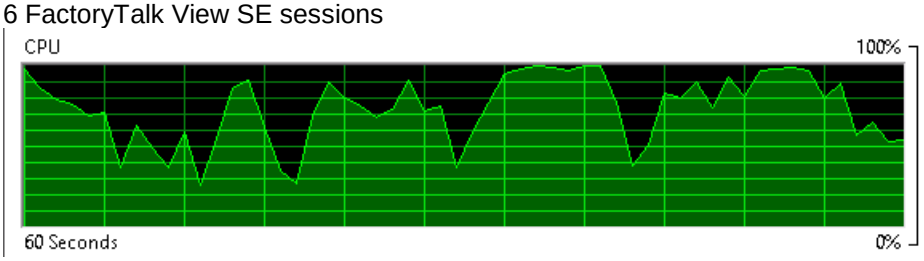
The ztC Edge performed quite well in phase 2 of testing, considering the application used could be viewed as unrealistically large for an OEM type solution. Noticeable delays when navigating overview screens or calling up faceplates began occurring with variable frequency with 8 or more sessions running. We determined that a noticeable delay would be characterized by displays taking 2 or more seconds to render. The other metrics used in determining poor performance were the CPU and memory utilization of the RDS virtual machine as additional sessions were added. We characterized a sustained average CPU utilization above 75% to be the point of over-utilization and less reliability. Likewise we characterized sustained memory hard faults/sec above 100 to be a condition of over-utilization of the swap file – using hard disk space when RAM runs out. It would not be recommended to run a machine in production past these marks to ensure the peak performance and reliability expected of a production system. We found in this test that the limiting factor contributing to poor performance of the ztC Edge system was CPU availability. The stress placed on the system resulting in slow display navigation and faceplate rendering stemmed from CPU overloading. The graphics below display how the CPU and memory utilization gradually increased as additional sessions were established.

Phase 2 – RDS VM CPU Utilization

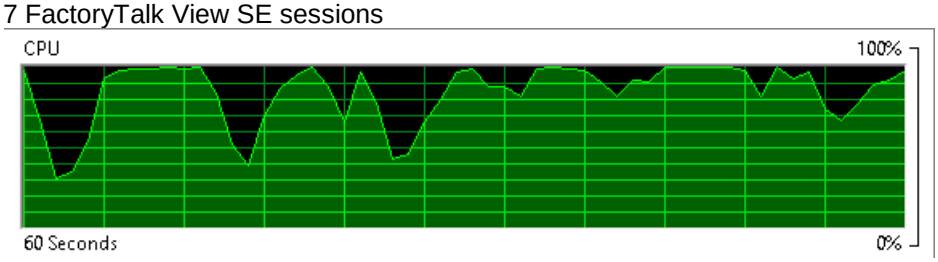
5 FactoryTalk View SE sessions



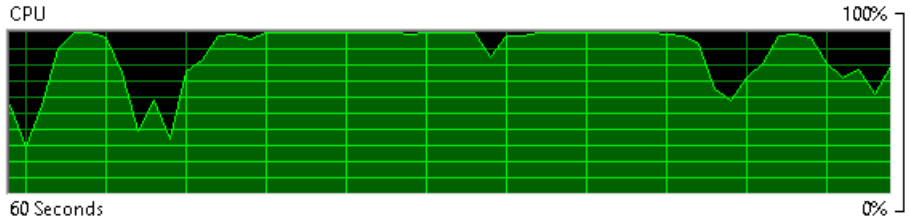
6 FactoryTalk View SE sessions



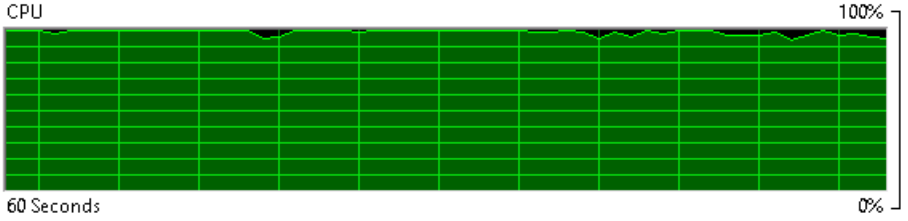
7 FactoryTalk View SE sessions



8 FactoryTalk View SE sessions



9 FactoryTalk View SE sessions

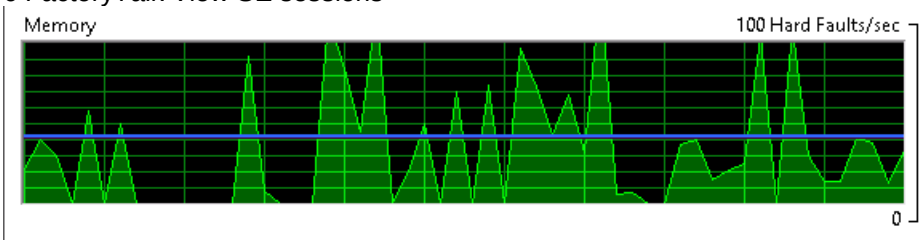


From the above graphics it is clear that the CPU utilization began to run above 75% on average after the 8th RDS session was established. This directly correlates to the point at which slower behavior of the FactoryTalk View SE Client application became visible to the user. Navigating overview screens and calling up faceplates began to variably take 2 seconds or more. The effect was not noticed on every page navigation, but every few navigation commands would result in one taking the 2 seconds or more. Once the 9th RDS session was established we began to see the lag in navigation very frequently. It is confirmed that this behavior is directly correlated to the sustained high CPU utilization at that number of sessions.

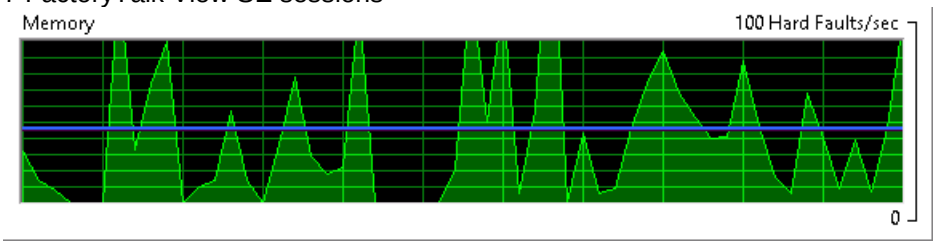
Phase 2 – RDS VM Memory Utilization

5 FactoryTalk View SE sessions

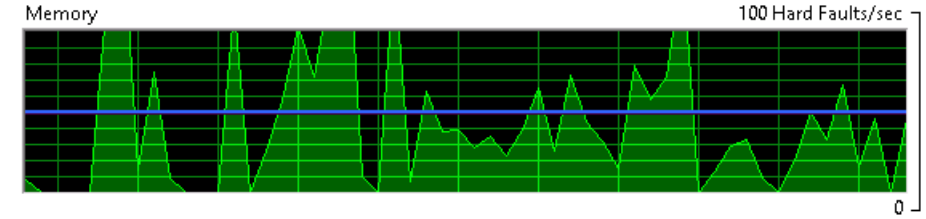
6 FactoryTalk View SE sessions



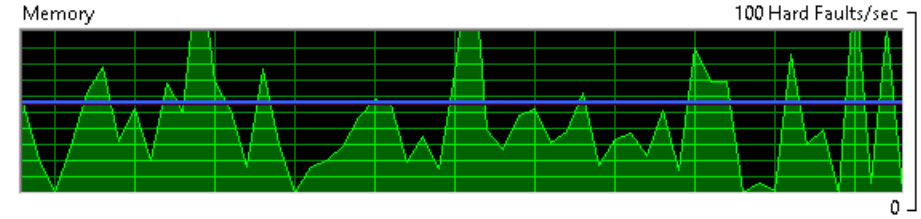
7 FactoryTalk View SE sessions



8 FactoryTalk View SE sessions



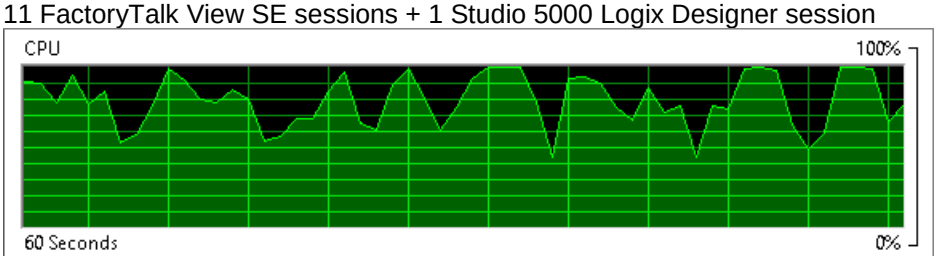
9 FactoryTalk View SE sessions



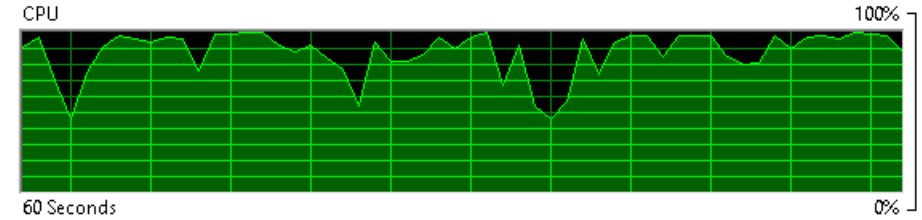
From the above graphics it is clear that the memory utilization was not a limiting factor in these tests. The memory allocation to the RDS virtual machine was adequate to handle the loading associated with 9 RDS sessions.

Phase 1 – RDS VM CPU Utilization

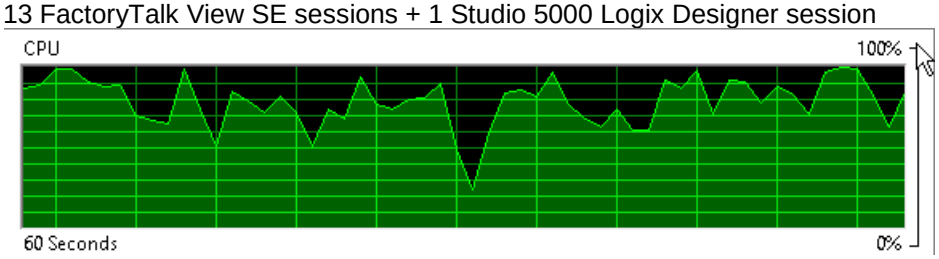
11 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



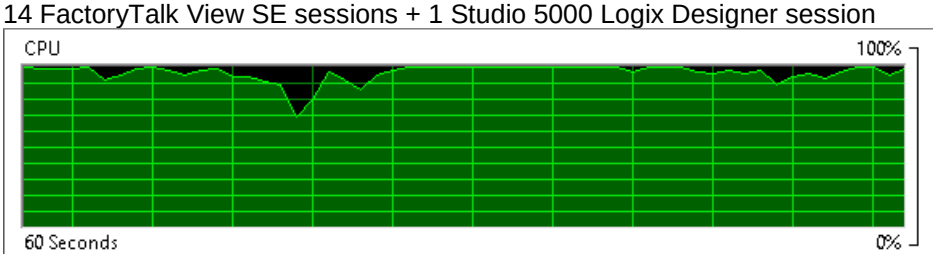
12 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



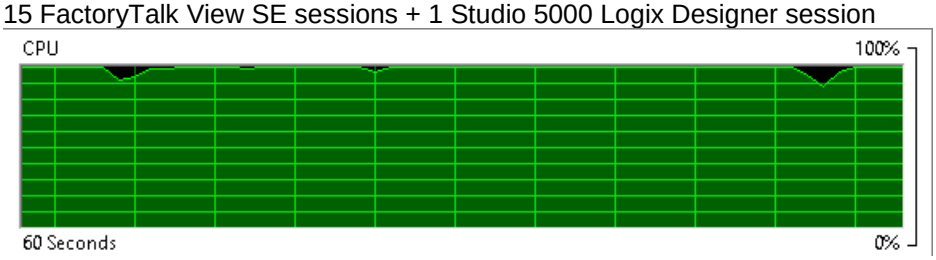
13 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



14 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



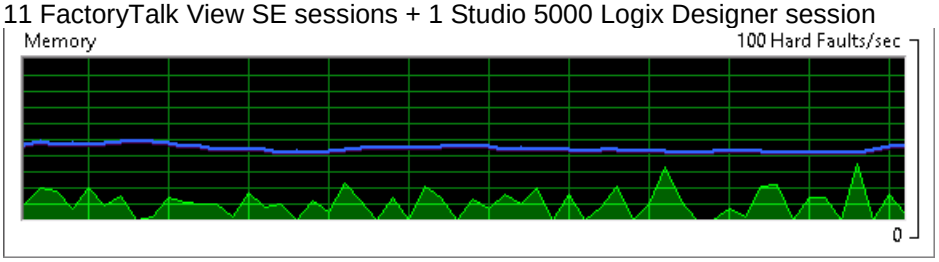
15 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



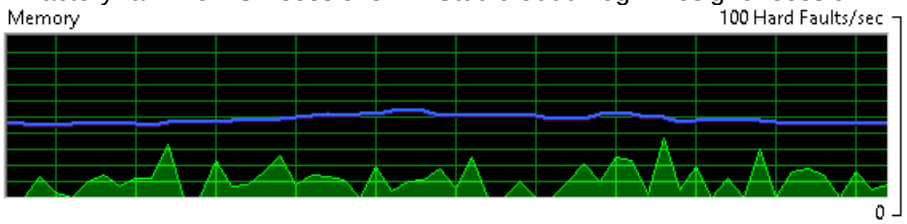
The above graphics from Phase 1 testing show that the CPU utilization began to run above 75% on average after the 15th RDS session was established. This directly correlates to the point at which slower behavior of the FactoryTalk View SE Client application became visible to the user. Navigating overview screens began to variably take 2 seconds or more. The effect was not noticed on every page navigation, but every few navigation commands would result in one taking the 2 seconds or more. Once the 16th RDS session was established we began to see the lag in navigation very frequently. It is confirmed that this behavior is directly correlated to the sustained high CPU utilization at that number of sessions.

Phase 1 – RDS VM Memory Utilization

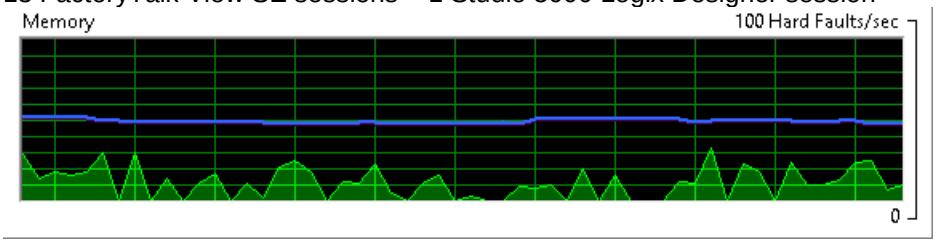
11 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



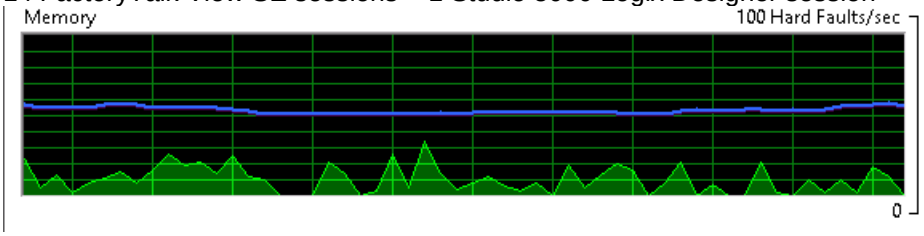
12 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



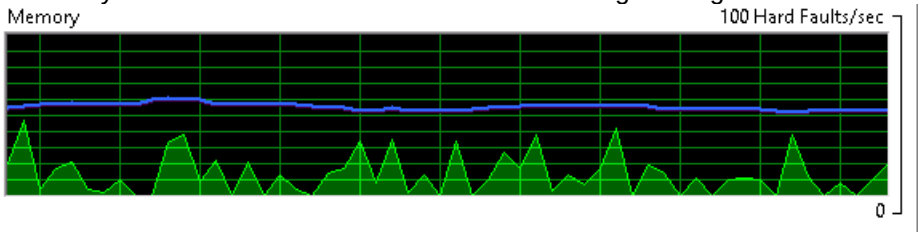
13 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



14 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



15 FactoryTalk View SE sessions + 1 Studio 5000 Logix Designer session



From the above graphics it is clear that the memory utilization was not a limiting factor in these tests. The memory allocation to the RDS virtual machine was adequate to handle the loading associated with 16 RDS sessions.