C1Po1F-04

Updating the Thermal Vacuum Chambers at the NASA Johnson Space Center

Cody Schaefer, Steven Del Papa, Jesus Macias

Introduction

SCADA systems for large cryogenics systems must seamlessly integrate complex control systems into a single robust and comprehensive system for users. When these SCADA systems approach their end of life, the process of updating them can present unique and time-consuming challenges. This presentation will cover the challenges faced when modernizing the SCADA system for Chambers A and B thermal vacuum systems at NASA's Johnson Space Center, along with the processes we implemented along the way.

Background

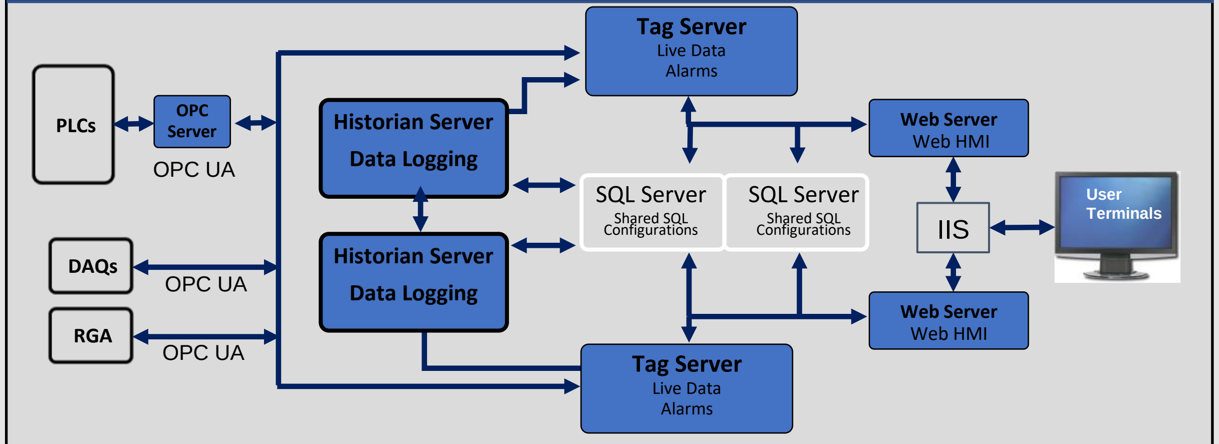
Chambers A and B are two large thermal vacuum chambers at Johnson Space Center which enable space simulation for unmanned and human-rated missions, respectively. With the resurgence in deep space missions for scientific research and the entry of various private commercial ventures, these chambers are expected to be used frequently for the next decade. The SCADA system for these chambers were using obsolete technology that could no longer be supported. The old system ran on Windows 2008 servers, required Active-X (Internet Explorer), had limited redundancy and interfaced using OPC DA. The process of resolving these problems required completely redeveloping the system re-interfacing with 17 PLCs and 12 DAQs, with tens of thousands of data points.

Goals

The main goal of this project was to redevelop the SCADA system implementing OPC UA connections, HTML 5, and updated servers. This had to be done in parallel to the legacy system as its will be continually used for testing. The new system architecture was designed with redundancy to limit future downtime and interruptions.

This process needed to be implemented efficiently and quickly to meet the schedule requirement. The sub-systems should be structured identically allowing the same webpages to be reused through out the system. Automation tools were also used whenever possible to make the development more efficient.

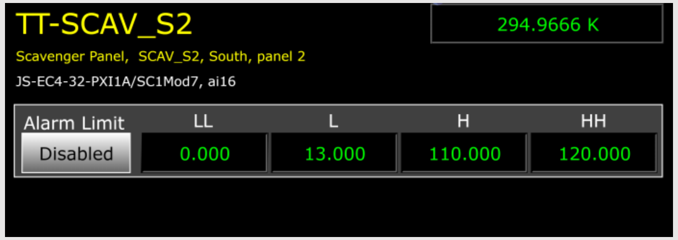
System Architecture



Results

The architecture of the redeveloped system is shown above with limited single points of failure compared to the previous design. The new design contains 3 pairs of redundant servers all interconnected with automatic failover between pairs and two SQL servers storing the configurations for all the servers. Each pair of servers serve a specific role with the first set of servers collecting live data over OPC, allowing live control, and monitoring of all subsystems. The next pair are dedicated to the recording and storing of historical data at a rate of 1 Hz to a binary file. The last pair are webservers hosting load balanced webpages allowing users access to the system. All together this creates a system that can maintain functionally even when significant faults or malfunctions occur. Any single server in a pair can be completely rebooted with minimal interruptions to the system.

Scripting was a major contribution to the efficiency, accuracy, and speed required to develop this system. The legacy system already contained all the information required to create the tags and alarms for the new system but required major refactoring to be used in the new SQL configurations. VBA scripts were used to almost entirely automate the process of refactoring the old configurations to work with the new system. It also allowed us to standardize the structure of system tags. This uniform structure made the system easier to work with and comprehend, while also allowing standard pages to be used for nearly all data points in the system.



Conclusion

We were able to successfully modernize our SCADA system meeting all our requirements on schedule and several lessons were learned. We implemented OPC UA, HTML5, and updated to new servers meeting NASA IT requirements. The standardized tag structure we implemented was successful, reducing the total number of webpages by over 75%. The redundancy of the new system architecture has been effective for maintaining uptime even through major faults. We learned that this redundancy has come at a cost though, as the new architecture contains more servers than necessary increasing system complexity. Each additional server in the system adds another layer for potential problems, making troubleshooting and maintaining the system harder. An additional lesson we learned is that historical systems with proprietary file types should be avoided. The ability to query and store historical data outside of the SCADA software outweigh any benefit a proprietary file type brings.

Future Work

We are working on the development of a new system building off the lessons we learned during this project. The goals for this new system are to increase capability while reducing operational and hardware cost. We aim to do this by reducing the total number of servers in the system and moving away from a proprietary file type for our historical data. The current loads on our servers will allow us to reduce the total number of servers in our current system, reducing the total amount of hardware required for the project while also making troubleshooting and maintaining the system easier. Alternative SCADA software that offer direct access to our historical data is being explored. This is a requirement for us as we need to be able to directly query and store our historical data outside of our SCADA software.

Acknowledgements

Special thanks to the dedicated and talented engineering team who played an instrumental role in the design and implementation of this project.