There is a trend in Enterprise SCADA toward larger systems controlling more assets from a single location within the pipeline company. As additional control systems are required for new assets, expansion of existing assets or projects to replace legacy systems, it is natural to want to leverage or expand an existing successful SCADA infrastructure.

For example, we have worked with network operation centers (NOCs) that have integrated national crude, products and gas transportation networks under a single distributed SCADA infrastructure. In each case, the national systems began as local and then regional systems. Over time, more and more pipeline networks were brought under the control of the central operational authority, which expanded to encompass the national networks.

This centralization is due to many factors, including the simple efficiency of managing fewer distinct systems, and the proven track record of the respective SCADA organizations at providing secure, highly available and feature rich pipeline management systems to the enterprise.

These systems have become the linchpins of the pipeline company’s business, providing not only around-the-clock operational monitoring and control (including real-time operations, control room management and leak detection), but also applications for measurement and accounting, decision support and daily logistics.
The systems are critical both operationally and as the ultimate source of business information. Performance is a crucial factor, not only because these large and complex systems push harder on processing capacity, but also because larger systems control more assets, and the effect of slow performance has a wider effect operationally.

**Challenges In Verifying Performance**

Pipeline companies and SCADA vendors must ensure, before the system is put into production, that it will meet the operational needs of the business – including functionality and performance. There are three general factors that contribute to performance:

- **Load** – What is the operational load being placed on the system? (point counts, users, etc.)
- **Capacity** – What is the capacity of the hardware and network? (numbers of servers, numbers of cores, memory, bandwidth, etc.)
- **Output** – What is the measured performance of the system? (throughput, response time, use, etc.)

In a typical SCADA request for proposal (RFP), the load and performance requirements are specified in greater or lesser detail, and the vendor responds with the system capacity required to meet that load and performance. There is usually a requirement for a load test during factory acceptance testing, under both normal and heavy load conditions, to verify that the specified system meets the requirements before being shipped to site and put into production.

But specifying and then verifying performance of large SCADA systems is challenging. In part this is because every system is different, consisting of a heterogeneous mix of new and legacy devices, telecommunication networks, protocols, purpose-built applications and systems, IT infrastructure and interfaces.

The complexity of the SCADA environment makes the characterization of load difficult, and the team responsible for writing the specification may not know or be able to delineate the full behavior and load of the system to a level of detail that would allow an accurate simulation of the operational environment under operational conditions.

The simulation itself is difficult because of the limited extent that a large and complex environment of heterogeneous multi-vendor systems can be replicated – accurately or economically – in a factory environment. Adding to the complexity are requirements for growth, intended to ensure that the SCADA system can expand over its useful lifetime to include additional pipelines and stations.

In light of these problems, it is sometimes the practice to “over specify” the system, either on the front end by inflating the point count and operational load required, or in the procurement phase by specifying more computing power than may be needed to meet the nominal operational needs. At the lower end of SCADA sizes, from 20,000-100,000 telemetered variables, this often works. Adding
more or faster central processing unit (CPU) cores and physical memory will result in the ability to handle larger loads in almost linear fashion.

But for large and complex systems, increasing gross computing capacity beyond a certain point will have almost no effect on performance; you will merely increase the percentage time the CPU is idle without getting more work done per unit time. This is because the notion of “capacity” in large complex systems must also take into consideration the inherent efficiency of the network of software components to perform operations in parallel and efficiently control access to shared resources.

We suggest a process and technological framework for characterizing and managing system performance that can be delineated by:

• Specifying load and performance requirements with as much contextual information as possible.
• Designing the system for both deterministic performance and efficient use of resources.
• Validating and tuning performance during factory tests designed to exercise the system in as close to production environment as possible.
• Measuring performance to capture the specific state of the SCADA system, not merely gross performance metrics, such as CPU, random access memory (RAM), input/output (I/O) and network use.
• Maintaining and monitoring performance in production as it expands over its lifecycle.

**Load, Performance Needs**

It is common practice in SCADA RFPs to include both the expected load at time of system installation, plus a growth factor to accommodate expansion over the system’s effective lifetime. When specifying load and expected performance, it is useful to characterize as accurately as possible the real conditions within which the system will exist. Some load typical parameters include:

• Number of RTUs
• Number of telemetered variables divided into counts of analog vs. digital, input vs. output
• Number and type of communication circuits
• Historical samples collected per unit time
• Time window of historical information to keep online
• Number of operational consoles

System expansion requirements are often expressed in terms of additional point count. Spare runtime capacity requirements are often expressed in terms of available CPU and memory during the factory load testing. However, spare CPU and RAM may bear little relation to actual spare capacity.
Data Acquisition

One challenge in characterizing and validating data acquisition performance is in accurately accounting for the variety of field systems. Specific protocols and communication links can greatly affect perceived performance. For example, Modbus RTU over a high-latency very small aperture terminal (VSAT) channel will perform poorly regardless of the bandwidth available, due to its simple poll-response behavior.

These factors are typically accounted for and avoided in the design of the telecommunication system itself. However, in general, it is valuable to know the specifics about what protocols (and protocol operating modes) are being used, in order to properly test SCADA performance. An example would be SCADA hosts that receive reports by exception from sub-systems, where the rate at which exception reports come in should be accounted for as system-specific factor in test design.

A final example is the telecommunications infrastructure and how it operates. For instance, understanding the effect of SCADA cold start and failover connection behavior on cell-based IP communication hubs, where a sudden flood of TCP socket connection requests could affect the intermediate telecommunication system. The more that is known ahead of time about such interactions, the better plans can be put in place to configure the SCADA host during the factory phase.

Measurement Data Acquisition

For gas pipelines, measurement information may be acquired, and possibly over the same telemetry network as SCADA information. Some useful parameters to consider when specifying system size are: measurement end points, flow runs per measurement point, gas quality points, AGA parameters uploaded, flow computer events generated and the number of hourly vs. daily measurement points.

Performance tests for measurement applications should focus on the bursts of activity that occur at the top of the hour and the top of the day.

Once the real-time, historical, and measurement values are acquired, performance concerns shift from data acquisition and processing to data storage and access. Relevant factors to consider include how much data is kept online, and how often and how much of it is accessed by how many users.

It is important to distinguish between data stored for online use in the control room and data stored for long-term access from a decision support site. It is common practice to keep a smaller high-performance set of historical values (events, time series) online in the control room for dedicated access by controllers for a one- or two-month window, and then keep years of information online on a decision support system accessible from the corporate network, for engineering analysis, planning and regulatory reporting. The relevant data retention amounts should be specified for each site.
For user access, some metrics that should be considered for system performance include number of:

- controller positions
- displays open and called up per minute
- dynamic values on each display
- commands per minute (setpoint, valve/pump/compressor control, etc.)
- alarms arriving for acknowledgement
- active alarm conditions (acknowledged and unacknowledged)
- trends called up
- pens and data samples shown per trend
- reports run
- dynamic information on each report

As with the distinction between historical data retention inside and outside the control room, the performance specification should differentiate remote graphical user interface (GUI) sessions that connect to the control room vs. “casual user” sessions that access a replica of the real-time and historical data on a decision support site.

As mentioned, systems are sometimes over-specified from a hardware standpoint, as insurance against uncertainty about performance requirements. But for large systems, it is often factors other than CPU, RAM, and IO that limit performance. An important distinction here is between average performance vs. peak-load performance, something that often differentiates management of SCADA systems from typical IT systems. SCADA demands deterministic performance.

A source of anxiety in some SCADA departments is that IT-managed infrastructure will be optimized for use, at the cost of deterministic response times. For example, virtualized infrastructures may result in smaller footprints and higher use of available servers, but with less predictability in system throughput and response time. For critical infrastructure SCADA, predictable response time is an important safety feature.

So while efficiencies are certainly possible, they must be carefully managed, particularly on the production servers and control room workstations. CPU cores, memory, and IO cards should be assigned to specific SCADA functions to guarantee computing capacity and deterministic performance. Storage area networks (SANs) should be sized and tuned for fast insertion of and access to operational data (instead of optimizing for use of available disk capacity). Database design should carefully consider how the data will be inserted and then used by control room and business users to ensure the most effective design of indices and storage layouts.
On the SCADA design side, deadbands should be tuned to reduce data processing, storage, and transmission requirements. Data acquisition schedules and strategies should be reviewed to ensure that poll rates reflect what is actually required. This can include careful design of remote scheduling of measurement uploads, poll rates and minimum poll cycle times, fast scanning of feedback ranges after commands are issued, etc.

One design consideration is whether to partition the larger system into a distributed network of smaller systems or to specify a single system capable of handling the full operational load by itself. We have had success with both approaches, and much depends on customer preference for managing systems and the need to integrate information from multiple systems “at the glass” (i.e., on individual user’s workstations or reports). One compromise is to have multiple SCADA systems feeding a single decision support system.

With a sound system architecture and design informed by a complete and accurate view of system load, the next step is to validate and tune performance in the factory before the system is put into production.

**Validation, Tuning**

Factory validation of SCADA performance can be challenging, as the disparate real-world systems the SCADA system will interact with cannot be fully replicated in a test environment. Performance testing is ideally done in two phases: validation of core product performance at levels commensurate with project requirements; and validation of project-specific performance, using simulators.

The product-level testing establishes the system sizes and loads that could be sustained with acceptable throughput and response times, given a particular hardware configuration. These tests make industry-specific assumptions about the processing and operational characteristics of the SCADA system, as well as typical management and administrative activities.

For example, oil and gas pipelines, while consisting of many similar base elements, have different profiles in terms of polling and data update requirements. Gas systems may require more data acquisition of measurement-related information within the same telemetry network as the SCADA system while oil systems require more aggressive poll rates because of the need for timely tracking of batched product movement. With accurate product-level validation of performance in place, the project can then confirm that performance aligns with their specific requirements.

The system is put under load in both typical and heavy load configurations. Simulation is typically used to put load on the system, in a manner similar to what it will be subjected to in the field. For example, automated test harnesses can be used to play back realistic data and simulate the existence of thousands of field devices. For GUI performance testing, automated scripts produce load similar to real user activity.
Slow or unreliable telemetry networks are simulated with bandwidth limiters on the test LAN and protocol simulators can produce fake noise and simulate other failures. The full system is tested, including the main SCADA site, backup site and decision support systems.

The intent of the heavy load test is to push the system beyond its normal limits. Performance tuning in this phase is done to ensure that the system can maintain acceptable levels of throughput and response time, even under stress. Large system sizes present additional problems not necessarily noticed on smaller systems.

In this phase, the symptoms of performance issues may not actually reflect where in the system the bottleneck exists. It is necessary to “peel the onion” – finding and eliminating one bottleneck, which then exposes another bottleneck that must be identified and eliminated, and so forth. The tuning process for large systems is inherently iterative and may require access to SCADA performance counters, such as data processing queue sizes, transaction activity and cycle times.

With the combination of the insight provided by internal and external performance numbers, and a process of finding and tuning performance issues, system performance can be improved by orders of magnitude – from tens of thousands, to hundreds of thousands, to millions of telemetered variables.

**Measuring Performance**
Throughout testing and then into production, it is necessary to measure performance. Typical metrics include CPU and RAM use, network bandwidth and run queue sizes. For large systems, these measures may not be enough, as the high amount of parallel activity and many users produce bottlenecks that are a function of parallelism and shared resource access, rather than raw computing power.

In such cases, the internal metrics of the network of interacting components of the SCADA system begin to play a more important role in analyzing and tuning performance. Such metrics can include replication and telemetered data processing queue sizes, transaction times, lock wait times, etc. By exposing these metrics in a standard way, a more holistic view of SCADA performance can be obtained.

From product and factory-acceptance testing, a tuned system is delivered that can handle the operational loads expected of it. The system must then be continually monitored to ensure that it continues to accommodate real-world loads, both at system installation time and over its operational life.

**Conclusion**
There is a growing requirement for large pipeline SCADA systems and performance is becoming a crucial aspect of system design and operations. Typically, performance has not been adequately
characterized in the specification phase, and has been difficult to validate during factory-acceptance testing. By more accurately describing system load, validating performance using higher-fidelity models during factory testing, and leveraging a SCADA health-monitoring infrastructure that integrates both SCADA and IT performance metrics, large systems can be delivered and managed with much higher confidence.

*Editor’s note: This article is based on a presentation at the ENTELEC 2014 Spring Program held in Houston.*

**Author: Kevin Mackie** is from Calgary, Alberta, Canada. He has a B.Sc. in Computer Science from the University of Calgary and has worked in the Oil and Gas Pipeline SCADA Industry since 1992. He is currently a SCADA Product Manager in Schneider Electric’s Oil and Gas Segment.