LEARNING BRIEF

Demystifying Industrial Data Acquisition

BACKGROUND

Supervisory control and data acquisition, or SCADA, is a computer system for gathering and analyzing real time data. SCADA systems have come into prevalence in their support of telemetry data and industrial automation. Historically, these systems have been relegated to field operators tasked with maintaining operations. Consequentially, analysts that want to better explore this data must use archaic software applications built more for day-to-day management than data exploration. In response, technology teams have closed the gap by batching files of summarized operational data for inclusion into a data warehouse, but analysts are beginning to see the value of (and asking for) the raw data. Unfortunately, data access for industrial systems can appear opaque at first glance. What follows are 3 of the core principles spanning the architecture, sourcing, and publishing of data to demystify scaling up a solution quickly.

Standards & Architecture Basics

Getting your hands on data is a matter of standards from the Open Platform Communications (OPC) Foundation. These standards enable SCADA vendors to unify communications among a wide range of device manufacturers. From the data engineer's perspective, the entire system is a classic client/server Figure 1 - Basic Architecture architecture, so the primary task is to implement a client that can interface securely with a vendor's OPC server. The server acts as a hub for connected devices, and vendors that implement OPC Data Access (DA) will typically offer a Software Development Kit (SDK) to tap into real-time communication. However, the DA standard does not require the vendor to keep history. Vendors that commit to OPC Historical Data Access (HDA) or Unified Architecture (UA) will keep history and typically offer an ODBC driver for database access. Best-in-class vendors will integrate with cloud platforms like Microsoft Azure, so you don't have to build custom software.

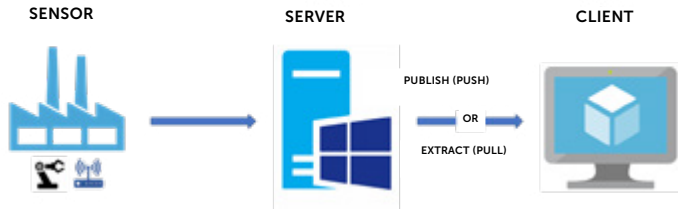

FIGURE 1 - BASIC ARCHITECTURE

Considerations On Push vs. Pull Sourcing

A data architecture has two fundamental data sourcing modes: the server can *publish* data, or the client can reach in and *extract* data. A publishing process will *push* data to external systems as server data becomes available.

In contrast, extracting data means that a scheduled query will *pull* data from the server. A push architecture tends to exhibit stronger data quality and performance characteristics because processes are close to source data infrastructure and designed—ideally by the vendor—to emit events without business logic.

Publishing For Analytics

Servers can publish sensor data in the form of *messages*; implement database replication; or write directly to external systems like Azure Data Lake Storage (ADLS). Streaming messages to a *broker* like Azure Event Hubs will result in the lowest possible latency because the server emits sensor event messages as fast as it can durably flush them from memory. Database replication will also have low latency, but the server may have encoded messages into a dense schema of tables, so reconstituting the original event may be difficult. Finally, writing directly to your preferred database system is a boon for productivity, but data won't stream to a target; instead, the system may batch writes no faster than every 5-15 minutes. A data extract query can be scheduled with a similar latency, but it will not be event-driven, instead relying on a high-water mark or similar process.

Parting Thoughts

We believe that streaming can be the backbone of your data platform even if you don't have a use case today. CCG recommends clients: 1) work with cloud platforms like Azure which support HDA or UA architecture, 2) strongly consider push methods in transmitting data to ensure best possible data quality, and 3) take advantage of event hubs. Choose a straightforward, vendor integration first. If you need a custom solution, make sure your team can realistically maintain a software application. In the end, data quality will matter more than performance.