influxdata®

AN INFLUXDATA TECHNICAL PAPER

Industrial IoT with InfluxDB

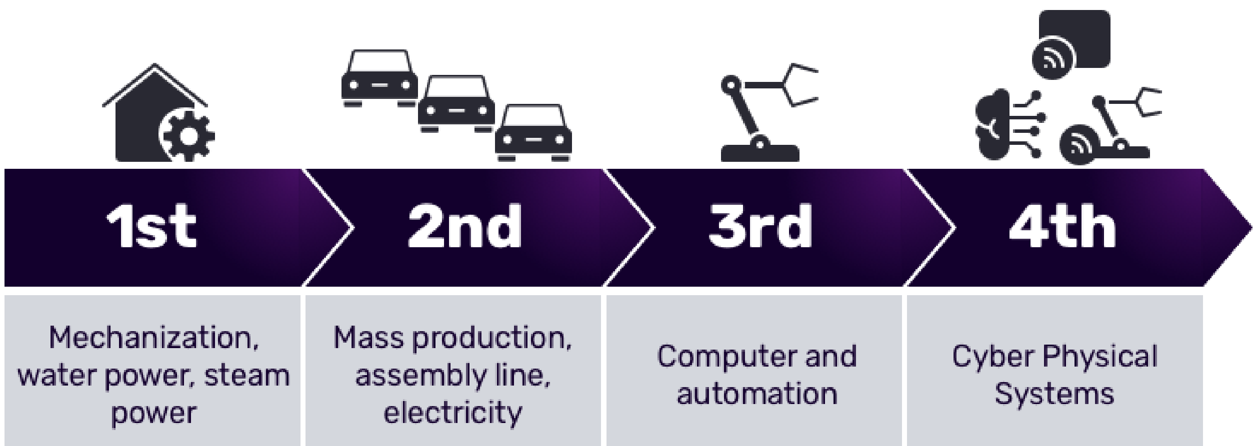
FEBRUARY 2021 - REVISION 2

Introduction

The industrial world has a long history of modernizing its process controls in order to keep production running efficiently and safely while minimizing downtime. Yet many are locked in established data historian solutions that are costly and lack the methods needed to provide innovation and interoperability. In contrast, open source software — which is built on the foundation of community — inherently provides diverse design perspectives not available from a single software vendor. It provides freedom from vendor lock-ins, which means it will always enable you to integrate with other solutions. Furthermore, open source software provides customization, allowing you to adapt the code to fit your ever-changing system requirements (which is not easy with proprietary systems). In this paper, we will examine what existing solutions lack and review a few open source projects that should be considered for future success for operators.

Industry 4.0

Industry 4.0 is the term used to describe the 4th movement in the evolution of manufacturing. Starting with the Industrial Revolution (water & steam power), the introduction of the assembly line, then followed by the adoption of computers and automation (robotics) — these were movements that disrupted the status quo and changed how things were made with each step. The next phase, Industry 4.0, is another evolutionary movement to enhance manufacturing processes via autonomous systems being fed raw and trained data (machine learning).



The design principles behind Industry 4.0 are simple and make it compelling and worth considering in the context of time series data:

- **Interconnection** — The ability to have devices, sensors, and people connect and communicate with each other.
- **Information transparency** — Interconnection allows for the collection of large amounts of data from all points of the manufacturing process. Making this data available to the operators provides them with the informed understanding that can aid in the identification of areas of innovation and improvement.
- **Technical assistance** — The ability to aggregate and visualize the data collected with a centralized dashboard allows operators to make informed decisions and solve urgent issues on the fly. Furthermore, centralized data views help operators avoid conducting a range of tasks that are unpleasant or unsafe for them to perform.
- **Decentralized decisions** — The ability for systems to perform their tasks autonomously based on data collected, and only on an exception basis, require interference.

These principles make sense, but how do you go from where you are now to this new level of manufacturing?

Current situation

Industrial organizations around the world, whether large or small, have been working with a number of solutions to digitally transform their manufacturing processes. Most organizations use a system of software and hardware components called Supervisory Control and Data Acquisition (SCADA) to help control machinery and systems in a factory in real time. In particular, these systems control processes locally by gathering and recording event data from sensors, valves, pumps and motors. In addition, the relevant data is presented to the operator locally to make decisions about the machinery to keep it running optimally. Many industries rely on SCADA systems — including energy producers, manufacturing, and food and beverage — to collect event data such as:

- **Instrument readings** (flow rate, valve position, temperature)
- **Performance monitoring** (units/hour, machine utilization vs. capacity, scheduled vs. unscheduled outages)
- **Environmental readings** (weather, atmospheric conditions, groundwater contamination)
- **Production status** (machine up/down, downtime reason tracking)

All process and event data includes a value and a timestamp and is stored in a data historian to show trends per machine or across a collection of machines. A data historian is essentially a time series database, and as such, needs to allow for fast ingestion and query of data in near real-time and provide compression of the data to minimize storage. There are many commercial data historian solutions in the market, and several have been in the market for some time, yet all these solutions come with a number of challenges — primarily cost, vendor lock-in, and scalability.

Cost — These solutions are not cheap, charge an annual license and support fees and are costly to setup and maintain. Moreover, custom development on top of these off-the-shelf products is common, which may require outside consulting resources. And since these solutions are proprietary systems, the work is time-consuming and expensive.

Vendor lock-in — These solutions are often Windows-based and do not offer a simple, open API for other software to interface with. This means you are limited to integrate and buy all components from only one vendor, locking you into a proprietary solution.

Scalability — Collecting event data from your equipment is just the beginning. True digital transformation requires more data sources and more analysis of the combined data to gain a better understanding of your systems. Doing so with existing solutions will require vendors to create (and charge for) new interfaces for data import. The good news is that this data is easy to export to spreadsheets. The bad news is that spreadsheets only give you a static view. What is required instead are modern dashboarding engines which obsolete the idea of exporting large time series data sets. Ultimately, with all this data coming in, you can no longer rely on manual techniques for analysis.

The time series workload

Now that we have established that IoT data is time-stamped data and that data historians are essentially time series databases, let's examine the characteristics of time series data. Time series data has a few properties that make it very different from other data workloads. Data lifecycle management, summarization and large range scans of many records are what separate time series from other database use cases.

With time series, it's common to request a summary of a larger period of time. This requires going over a range of data points to perform some computation, like a percentile, to get a summary of the underlying series to the user. This kind of workload is very difficult to optimize for a distributed key value store. InfluxDB is optimized for exactly this use case giving millisecond-level query times over months of data.

With time series, it's common to keep high-precision data around for a short period of time. This data is aggregated and downsampled into longer-term trend data. This means that every data point that goes into the database will have to be deleted after its period of time is up.

This kind of data lifecycle management is difficult for application developers to implement on top of regular databases. They must devise schemes for cheaply evicting large sets of data and constantly summarizing that data at scale.

An open source alternative

To contend with these challenges, the industrial sector should consider new ways for optimizing operations, including trying open source. Previously, open source held the stigma of being the cheap alternative to proprietary software. Today, open source is at the heart of innovation in organizations, as it allows developer teams to quickly bring ideas to fruition faster.

With the availability of so many open source tools, operators are no longer required to purchase arcane closed source solutions. Instead, they can look to building their own data historian replacements or buy a ready-made solution that is based on open source. This provides the operator with the freedom to quickly innovate and never be locked in to a single solution that could easily and quickly become obsolete.

InfluxDB & Telegraf

InfluxData is the creator of InfluxDB and Telegraf — two open source projects with vibrant communities that help bring time series data from a number of sources into a purpose-built time series database. Built for developers, InfluxDB is central to many IIoT data historian solutions providing high throughput ingestion, compression and real-time querying of that same data. Efficiency and effectiveness have to start in the data structure, ensuring that time-stamped values are collected with the necessary precision and metadata to provide flexibility and speed in graphing, querying and alerting. The InfluxDB data model takes the following form:

```
<measurement name>,<tag set> <field set> <timestamp>
```

The measurement name is a string, the tag set is a collection of key/value pairs where all values are strings, and the field set is a collection of key/value pairs where the values can be int64, float64, bool, or string. Support for data encoding beyond float64 values means that metadata can be collected along with the time series, and not limited to only numeric values (e.g. valve is open or closed). In addition, there are no hard limits on the number of tags and fields.

Having multiple fields and tags under the same measurement optimizes the transmission of the data, which is important for remote devices sending metrics. The measurement name and tag sets are kept in an inverted index which makes lookups for specific series very fast. See below how you could ingest battery metrics into InfluxDB with:

```
telemetry,product_id=1,battery_type=lead-acid voltage=10.1,current=-0.5,temperature=23.4 1464623548s
```

Telegraf is the collector agent that boasts more than 200 Telegraf plugins with the majority provided by the open source community. It is written in Go and compiles into a single binary with no external dependencies, and requires a very minimal memory footprint. The variety of plugins created with Telegraf include gathering data from:

- Databases: Connect to data sources like MongoDB, MySQL, Redis and others to collect and send metrics.
- Systems: Collect metrics from your modern stack of cloud platforms, containers and orchestrators.
- IoT sensors: Collect critical stateful data (pressure levels, temp levels, etc.) from IoT sensors and devices.

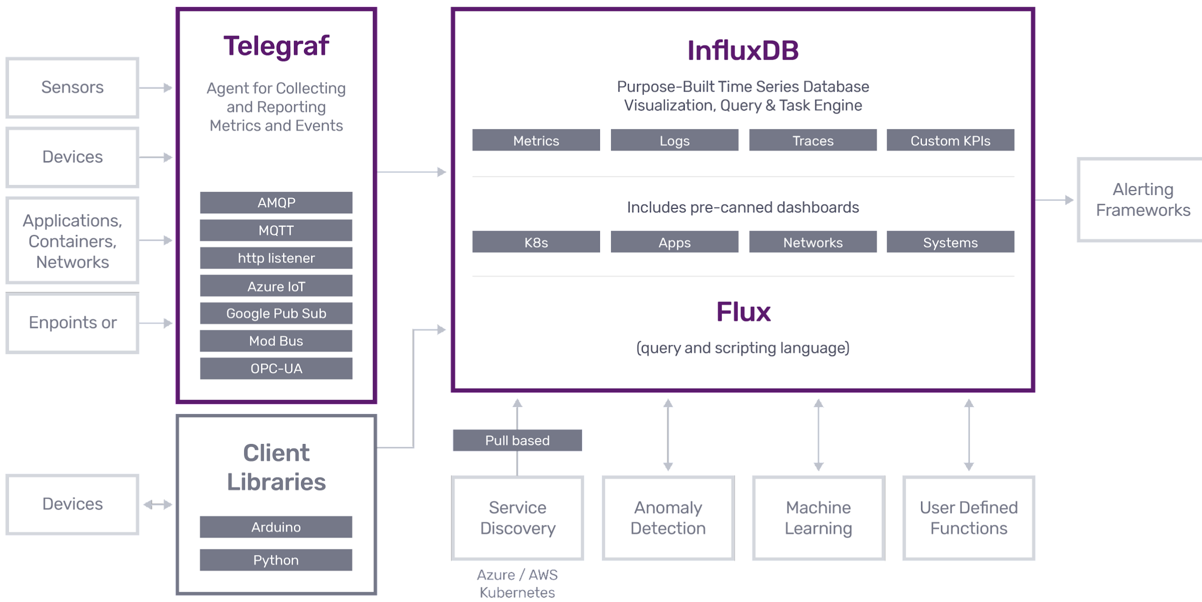
Some of the IIoT specific Telegraf plugins include:

- AMQP Telegraf Plugin — The AMQP Consumer Telegraf Plugin provides consumers the ability to receive streaming data through an AMQP 0-9-1 compatible broker, like the aforementioned RabbitMQ. The metrics are read from a topic exchange using the queue binding_key.
- ModBus Telegraf Plugin — The Modbus Telegraf Plugin collects Discrete Inputs, Coils, Input Registers and Holding Registers via Modbus TCP or Modbus RTU/ASCII. In the configuration setting, you can identify the addresses of the Modbus device on the bus, the range, timeouts, retries, etc. to collect the measurements directly from the equipment, from the SCADA, or from the automation systems.
- MQTT Telegraf Plugin — MQTT pulls in all your time series data (metrics and events) from your applications, infrastructure, and even sensors, making it easy for your developers to use. The MQTT Consumer Telegraf Input Plugin reads from specified MQTT topics and adds messages to InfluxDB. Messages are in the Telegraf input data formats. You can gather and graph metrics from your IoT devices with the Message Queue Telemetry Transport (MQTT) protocol — a simple and lightweight messaging protocol ideal for IoT devices.
- OPC-UA Telegraf Plugin — OPC Unified Architecture (OPC UA) is a communication protocol for industrial automation developed by the OPC Foundation and released in 2006. The current version is 1.04 (released in 2017) and includes publish/subscribe and a client/server communications infrastructure. The OPC-UA Telegraf plugin helps you gather metrics from client devices using the OPC-UA protocol.
- Sensor Telegraf Plugin — The Sensor Telegraf Plugin can help you collect sensor metrics with any sensor executable from the Linux-monitoring (lm-sensor) package. lm_sensors is a free open source software tool for Linux that provides tools and drivers for monitoring temperatures, voltage, humidity and fans. Using this Sensor Telegraf Plugin — in conjunction with the many other Telegraf plugins like ModBus, MQTT, and OPC UA — can help you collect metrics from many types of sensors found in homes, factories, energy production sites, etc. Collecting all this data in a single time series database like InfluxDB can help you gain performance insights, predict maintenance cycles, as well as find optimization opportunities.

Bringing it all together

As mentioned, the InfluxData open source time series platform consists of InfluxDB and Telegraf, but there are many tools and protocols available with InfluxDB:

- Client Libraries — InfluxDB also supports a set of client libraries (Arduino, Java, JavaScript, Python, GoLang) for data ingest which can be used to write and read data to and from InfluxDB to any application or other data source.
- Visualizations — InfluxDB comes standard with integrations with popular dashboarding tools like Grafana and Seeq and also comes with a set of libraries called Giraffe that you can use to add visualizations into your custom application.
- InfluxDB Templates — Pre-canned InfluxDB Templates include all data ingestion configurations including the use of any Telegraf plugin, dashboards and alerting all compiled in an InfluxDB Template that you simply install in your InfluxDB instance to get started quickly. There are several IoT specific templates that include: Enviro+Blackberry Pi, Fireboard Monitoring Template, Island Pulse Monitoring Template (using ModBus), and several air quality tracking templates.
- Community-based plugins
    - APIs are available in several programming languages both as a commercial SDK and open source stacks. The open source stacks include language support for C, C#, C++, JavaScript, GoLang, Python, Java and Rust.
    - OPC-UA stack, written by the team at Factry.IO, polls for the values on the PLCs then writes them directly to InfluxDB.
    - OPC-UA Server support — With InfluxDB, you can collect and analyze process inputs and outputs, and suggest optimized setpoints to the SCADA system with the OPC-UA server capability. Examples include using neural networks to analyze data or incorporating weather forecast data into the control mechanisms.
    - Ockam ExecD Plugin — This plugin provides end-to-end encrypted connections between Telegraf and InfluxDB.



Case study: Build a data historian (Factry.IO)

Factry brings open source software to the world of manufacturing with their solution, Factry Historian, a data collection platform for production systems. They use InfluxDB to store time series data and have open-sourced two projects to collect and expose time series data using the OPC-UA standard. The OPCUA-to-InfluxDB open source app polls a number of PLC values and writes the data into the database. If the database is unavailable, it buffers the data in a local database. You can see the types of values they are able to collect and store.

```
[[measurements]]
name              = "temperature"
tags              = { equipment = "TANK42" }
nodeId            = "ns=3;s=PLC_TANKS.db103.16,r"
collectionType    = "polled"
pollRate          = 12    # samples / minute.
deadbandAbsolute  = 0     # Absolute max difference for a value not to be collected
deadbandRelative  = 0.0   # Relative max difference for a value not to be collected
```

They also open-sourced an OPC-UA server that exposes the data stored so a SCADA system can connect to this server and read the data at the interval it needs.

The Factry solution can serve as a model for creating a data historian replacement, using open source projects for data collection and storage. Furthermore, if the operator chooses to buy a solution like the one Factry has created, with open source as the base, there is little risk of vendor lock-in as the data can be easily transferred to another solution.

Case study: Moxie IoT

MOXIE IoT uses industry standards and cutting-edge technology to create a monitoring solution which provides their manufacturing customers with a single source of truth. Their iOS iPad app equips their customers to visualize and analyze industrial factory data in real time, essentially a Digital Twin. The MOXIE IoT tracks the movement and activity of factory assets, and it aims to improve performance and safety. These assets include overhead cranes, forklifts, pallets and humans. MOXIE's moxieWORLD platform collects, stores and analyzes this data in an iOS app in real time and provides historical data. The platform provides customers with an intuitive overhead mapped view of its factories, etc. MOXIE IoT created its iOS solution using Python, Swift, MQTT, InfluxDB Cloud and AWS. The platform is built on InfluxDB and uses Flux to query its data.

Case study: Nortal

Nortal is a leading digital transformation company with 1,000 technology experts across North America, Europe and the Middle East. Nortal's team of data and analytics experts — paired with InfluxData's powerful cloud-based database, reporting and monitoring tools — equip their customers with faster time-to-market while ensuring elasticity, reliability, and scalability.

With over 30 years experience in implementing Data Historian solutions to create modern industrial platforms based on agile architecture principles, Nortal enables data-driven operations from the shop floor to the top floor. Partnering with InfluxData on Industrial IoT allows enterprises the freedom to manage the platform independently while ensuring speed, stability, redundancy, and scalability.

How time series helps with Industry 4.0

Now let's explore how time series data can help organizations transition to Industry 4.0 by reviewing an existing use case. This use case will follow a medium-sized textile plant that knows all too well that their competitive advantage fades as production costs increase. To circumvent this, they implemented an end-to-end solution that monitors their entire production process to realize a decentralized decision-making process and find new areas to optimize.

For this textile plant, customer orders include type of material, amount, and when the product becomes available for shipping. Of course, there are orders that come in simultaneously so they try to match production against these orders. A few things that they need to monitor and manage include; equipment (number of available machines, maintenance schedules, equipment failures), availability and management of dyes and raw materials, how much product can be produced from the raw materials, and staffing and scheduling. And finally, it is important to note that operators are still required to watch the machines to change or add raw materials, to check maintenance schedules, and watch for machine failures. Let's review the 4 principles that make up Industry 4.0 and how time series fits in to support this textile plant.

Interconnection

Each equipment already had controllers on them for the operators to control. And now that each piece of equipment is connected to a network, the collection of large amounts of time series data from all points of the manufacturing process can be sent to a centralized data store. This connects operators, supervisors, management, the equipment, and all the processes together.

Information transparency

Once connectivity is established and time series data is collected centrally, the data is made available to the operators in the form of dashboards to provide them with an informed understanding that can aid in the identification of areas of innovation and improvement. The dashboards are not static views of data often found in spreadsheets — they include real-time data and capabilities for the user to perform drill-down analysis at will.

Technical assistance

Since data from multiple data sources is now added to this centralized store, it is now easier for anyone in the plant to review the data. Personnel no longer need to download data to perform analysis. Furthermore, having this data always available allows each member to set thresholds that can trigger specific workloads.

Rather than having to wait all day, walking the floor to have to switch out raw materials, personnel can now work on other tasks and be notified automatically instead.

Decentralized decisions

The only way for systems or people to perform their tasks autonomously is to provide them with real-time data. We have already established that centralized dashboards with real-time data are available and have helped this plant accomplish this. Furthermore, they experienced firsthand that these dashboards allowed new personnel to come up to speed faster so they can quickly identify areas for improvement based on time series data.

Conclusion

The basic goals in Industry 4.0 of technological improvement and personnel empowerment can be accomplished by collecting and acting on time series data. Time series data is the only dataset that shows you the change in time of any value — temperatures, machine utilization vs. capacity, or even weather. It is central to making decisions at any point of a manufacturing process that can impact your business right away as it allows for an immediate reaction based on current circumstances.

About InfluxData

InfluxData is the creator of InfluxDB, the open source time series database. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by IoT devices, applications, networks, containers and computers. We are on a mission to help developers and organizations, such as Cisco, IBM, PayPal, and Tesla, store and analyze real-time data, empowering them to build transformative monitoring, analytics, and IoT applications quicker and to scale. InfluxData is headquartered in San Francisco with a workforce distributed throughout the U.S. and across Europe.

Learn more.

InfluxDB documentation, downloads & guides

Get InfluxDB

Try InfluxDB Cloud for Free

Get documentation

Additional tech papers

Join the InfluxDB community

influxdata®

Try InfluxDB

Get InfluxDB

Contact us for a personalized demo influxdata.com/get-influxdb/