DOI 10.5162/sensoren2024/P23

DisDAQ - A Modular and Distributed Measurement and Processing System for Industrial Process Optimization

Leander B. Hörmann, Christian Helmel, Veronika Putz, Christian Kastl
Linz Center of Mechatronics GmbH, Altenberger Str. 69, 4040 Linz, Austria
Email: leander.hoermann@lcm.at

Abstract

In the dynamic realm of industrial manufacturing, enhancing efficiency is important for optimizing resource use, minimizing operating costs, and boosting productivity. Addressing global competition, custom production demands, and sustainability goals necessitates improvements in machinery and facility efficiency and flexibility. These improvements require comprehensive information on machine states, environments, and processes, often achieved through enhanced data utilization and additional sensors. When enhancing existing sensor capabilities, in a first step, important key parameters like the required physical quantity, sensor sensitivity, sampling rate and clamping position need to be clarified. Since these parameters are often not known in advance, they need to be targeted in feasibility studies and initial measurement campaigns. This paper introduces a distributed data acquisition system, DisDAQ (Distributed Data Acquisition and Algorithms Quiver), designed for such feasibility studies and initial measurement campaigns as well as industrial process monitoring, analysis, and optimization. DisDAQ integrates data collection from a scalable array of spatially distributed sensors, supports various sensor types, and minimizes measurement campaign preparation. Leveraging the Robot Operating System (ROS) 2, DisDAQ employs containerized ROS2 nodes packaged with the package manager Helm for flexible, synchronized data acquisition and processing, facilitating edge computing to preprocess data. Three different real-world application scenarios are presented which show the flexibility and general applicability of DisDAQ for different application areas.

Keywords: Distributed data acquisition, measurement system, modular architecture, ROS, Kubernetes, industrial measurement campaigns, retrofitting

I. Introduction

In the ever-evolving landscape of industrial manufacturing, the need for improved efficiency remains a constant challenge, driven by the imperative to optimise resource utilisation, reduce operating costs, and increase overall productivity. As industries face the challenges of global competition, lot size one production and sustainability goals, the need to improve the efficiency and flexibility of machinery, facilities, and processes is becoming increasingly important.

Such an improvement needs additional information and knowledge about the state of the considered machines, their environment, and of the processes to which they belong. The additional information is typically acquired by leveraging existing machine data more effectively and – in most cases – by retrofitting additional sensors [1]. Industrial IoT sensors [2]-[4] are well suited for such retrofitting tasks.

However, before adding additional sensors to an existing plant, in a first step, important key parameters like the required physical quantity to be measured, sensor sensitivity and dynamics, sampling rate and clamping position need to be clarified. Since these parameters are often not known in advance, they need to be targeted in feasibility studies and initial measurement campaigns. To be as flexible as possible in terms of supporting different industries, machines, facilities, and processes, a modular and flexible data acquisition system which supports the management and the execution of such measurement campaigns with low integration effort and without major interferences of the production workflow is needed.

This paper presents a distributed data acquisition system engineered to redefine how industrial processes are monitored, analysed, and optimized. By seamlessly integrating raw data aggregation across a scalable set of spatially distributed sensors, this innovative system aims to reduce preparation effort for measurement campaigns, to provide a synchronized and reliable data acquisition, and to support various types of sensors (including analog and digital industry grade sensors as well as lab quality systems, cameras, …). Additionally, it allows capturing background information available from machine data. Especially when assessing the feasibility of machine learning methods, e.g., for

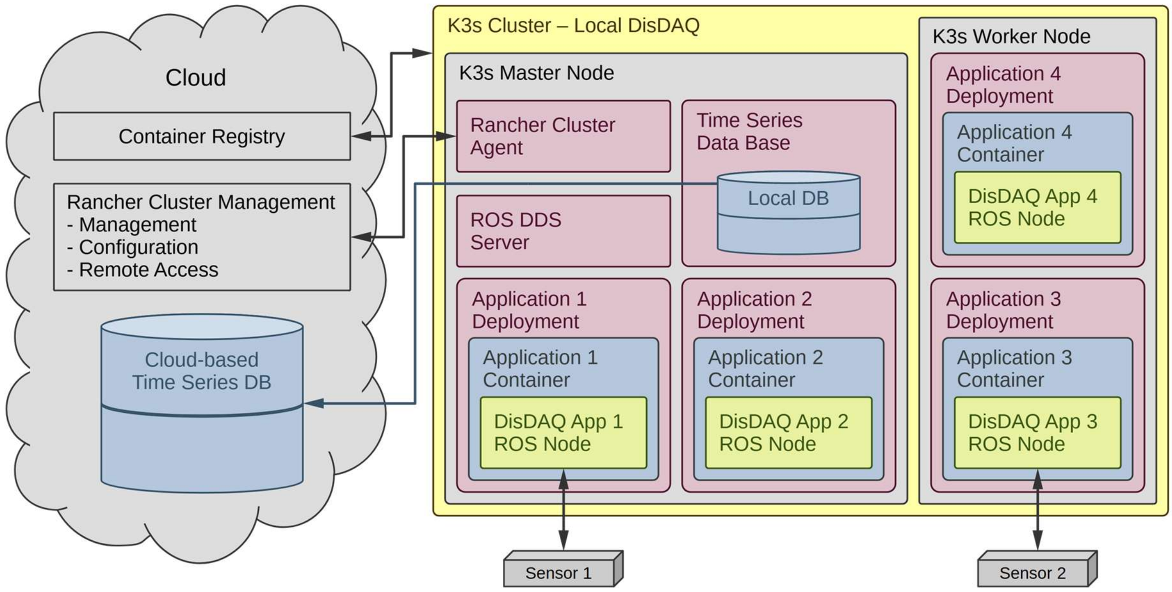

Figure 1: DisDAQ system architecture overview showing the K3s Kubernetes cluster (yellow) with one master node and one worker node. Each node has one or more deployments which are containerized. Inside these containers, the DisDAQ apps are executed which are responsible for interfacing the real sensors or data processing in real-time.

condition monitoring and predictive maintenance, the associated data-driven approaches require a highly increased amount of data which needs to be aggregated and analysed. To reduce the amount of data which needs to be transmitted, e.g., to a remote analysis system, preprocessing directly at the data acquisition system can be utilized. This, also known as edge computing, is also supported by the presented system. For the system at hand, the combination of a distributed data acquisition and the support of algorithm execution has led to its name DisDAQ (Distributed Data Acquisition and Algorithms Quiver).

It is based on the Robot Operating System (ROS) in version 2 as underlying communication and data exchange framework. Each ROS2 node of the DisDAQ system is responsible for a single data acquisition or data processing task and publishes the resulting data as ROS2 messages. The ROS2 nodes are containerized and can be individually parameterized and deployed in a local K3s Kubernetes cluster. Similar architectures have been presented but are not suitable to be run on resource constraint embedded pcs [5] or do not use a Kubernetes cluster for communication between parts of the system [6].

The presented approach enables a very flexible composition of the DisDAQ system for a certain measurement task. Since every ROS2 node of the DisDAQ system is part of the same ROS2 network, the nodes also share the same time base, which is necessary to enable a distributed data acquisition, combining data sources operating at different sample rates. Thus, measurement tasks can be started and stopped individually, and the recorded time-stamped data can easily be synchronized and resampled after recording. This allows for an easy combination of data from sensors employing their own internal clock (e.g., low-cost cameras) as well as sensors that can externally be triggered.

Beginning with data acquisition nodes which acquire data from physical sensors, a data acquisition and processing chain can be set up using further processing nodes. Finally, the resulting data will be used for analysis and visualization and is typically stored in a data base or transmitted to an upper-layer system.

The rest of the paper is organized as follows: Section II describes the system architecture of a typical DisDAQ system and its logical structure. Section III explains the typical steps to set up a measurement campaign and manage the system. Section IV presents three application scenarios which show the general applicability of DisDAQ for different application areas. Finally, Section V concludes the paper and outlines directions for future work.

II. System Architecture

The DisDAQ system architecture is shown in Figure 1 which is based on a K3s Kubernetes cluster set up on an Ubuntu server installation. The K3s Kubernetes cluster is a certified Kubernetes distribution built for IoT and edge device applications. The Kubernetes cluster consists of at least the K3s master node and optional K3s worker nodes depending on the application. Each of them represents a physical device such

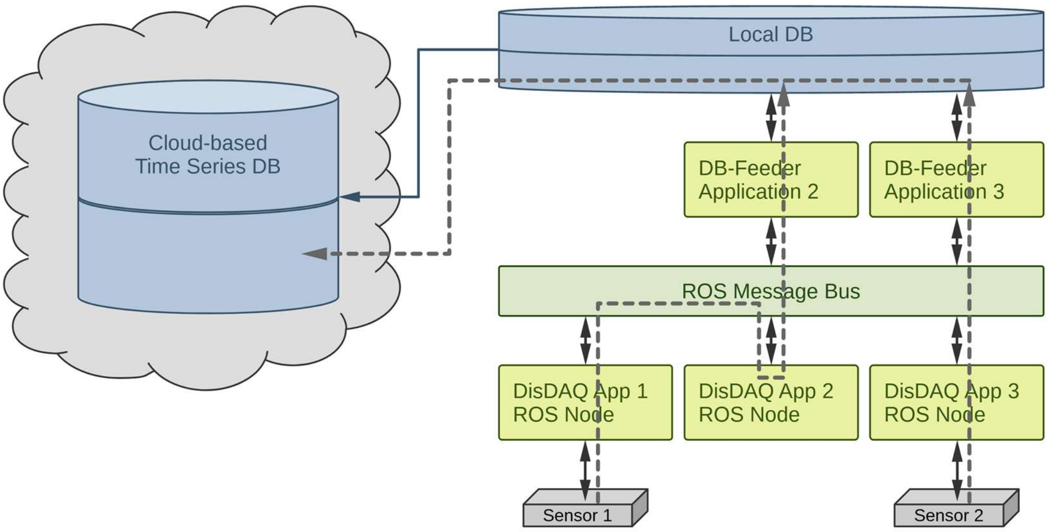

Figure 2: Logic structure of a DisDAQ instance showing the DisDAQ applications which are connected logically via the ROS message bus.

as an industrial PC or a single board computer (e.g. Raspberry Pi). These devices can be used also in distributed applications but are part of the local K3s cluster and communicate via its software defined network.
DisDAQ is deployed with the package manager Helm. It is used to package all necessary Kubernetes resources needed for the installation into a single, version-controlled package, the so-called Helm chart. The Helm chart defines the necessary cluster resources for the DisDAQ package to function as intended. All applications that are deployed for DisDAQ are containerized and packaged using Helm. A DisDAQ app package may contain drivers to interface sensors or algorithms for real time data manipulation or processing on the data served by sensor interfacing apps. The DisDAQ and DisDAQ app packages define all cluster resources that are needed for each app to function, such as local storage, network configuration, container arguments, cluster services and fine granular security context to add or remove capabilities for the application containers. Each DisDAQ app uses the robot operating system in version 2 (ROS2) to publish the resulting data. Thus, ROS2 is the underlying communication framework to connect the DisDAQ app and create a flexible data aggregation and data processing system. This logic structure is shown in Figure 2. It shows a DisDAQ instance using two physical sensors to gather data of physical quantities (e.g. temperature, vibration, images, …). Not every DisDAQ app is necessarily interfacing a physical sensor. As already mentioned, the DisDAQ app may also consume already gathered data and process them (down-sampling, statistical filtering, pattern recognition including image recognition…). This is also illustrated in Figure 2: Raw data from Sensor 1 are gathered using the DisDAQ App 1 and published via the ROS message bus. The data is used by the DisDAQ App 2 which is subscribed to the data from DisDAQ app 1. It processes the data and publishes the result via the ROS message bus. Finally, the data is subscribed by the DB-feeder DisDAQ App 2 which writes it into the local data base.
In Figure 2, the second aggregation example shows a more direct way: Raw data from Sensor 2 are gathered using the DisDAQ app 3 and published via the ROS message bus without preprocessing. The data is subscribed by the DB-feeder DisDAQ app 3 which writes it into the local data base.
Each DisDAQ instance (K3s DisDAQ device cluster) has typically a local time series data base (e.g., TimescaleDB or InfluxDB). Using a local data base has the advantage that its availability does not depend on network connectivity and bandwidth. Thus, it is well-suited to serve as primary storage solution. If necessary, the local data base can be configured to synchronize the local data with an online cloud-based time series data base. During periods with a stable internet connection, data is synchronized with the remote data base, where it can be accessed remotely as well. As alternative to using a local data base storage option, raw data can directly be saved as binary data files, which is especially helpful for aggregating raw data from large bandwidth sources (e.g., camera, lidar, imaging radar).
Besides the local data base, other optional plugins are available, only the most important ones are listed here:

- A live visualization tool allows for a brief status overview of the system. It can be used

DOI 10.5162/sensoren2024/P23

for verifying, if the system works as planned and supports debugging the system.
- The interactive development environment Jupyter Lab can be used for live data manipulation, which is useful during development of new functionalities or during the integration of new sensors in an operating system.

The DisDAQ instance (K3s DisDAQ device cluster) is typically managed via a cloud-based cluster management software (e.g. the open-source tool Rancher). Therefore, a cluster agent is deployed in the local K3s cluster. This cluster agent provides access for the cluster management software to the local Kubernetes API and therefore enables the cluster management software to view and edit the local cluster information and resources.

The novelty of this paper is the configuration for distributed in-cluster ROS2 nodes without the use of host networking, 3rd party software for networking, or the definition of extra VLANs.

ROS2 uses Data Distribution Service (DDS) and therefore uses a distributed discovery mechanism of ROS2 nodes by default. Without the use of VLANs and 3rd party SDNs (software defined networks), a centralized dynamic discovery mechanism is used. Therefore, a FastDDS Discovery Server is deployed in the cluster. The FastDDS implementation offers currently the lowest latency and second-best bandwidth performance compared to other DDS implementations according to [7]. Every ROS2 node container is configured to use the DDS Server via the Kubernetes deployment. This not only enables the possibility of distributed multi-node ROS2 communication within the K3s cluster but also reduces the network load for the discovery process by a factor of 15. To allow communication between all deployed ROS2 nodes and the dynamic utilization of ports, a K8s network policy is defined that allows all traffic within a 'ros' subdomain. All containerized ROS2 nodes reside in this subdomain. In addition, a headless service selects all ROS2 containers. To configure communication with the DDS server and to enable the use of the DDS server for all ROS2 nodes, a Fast DDS configuration defining a 'super_client_profile' is provided via a ConfigMap, and all containerized ROS2 nodes are configured to use it.

III. Measurement Campaign Management

Each measurement campaign using a DisDAQ system starts with setting up the hardware. Depending on the application, a suitable edge device (e.g., industrial pc) is selected as master node edge device. If necessary, further embedded computers are set up for distributed data acquisition and processing. The sensor

Figure 3: Hardware setup of a DisDAQ system inside a watertight housing for outdoor usage.

hardware is connected, and every component is mounted into a housing.

Depending on the application, watertight housing may be necessary. An example hardware setup for outdoor usage is shown in Figure 3. The hardware setup includes an embedded PC (black box on the right), three DAQs (light blue boxes in the middle), and the necessary cabling. Additional hardware for network connectivity and uninterruptible power supply, which allows controlled shutdown and notification in case of power interruption, are located at the lower level of the box (not visible in Figure 3).

For remote management of the DisDAQ installation, a new DisDAQ instance is set up in the cloud-based cluster management system. On the master node edge device, the host operating system and the DisDAQ packages are installed. To connect the local installation with the cluster management software a token obtained from it must be provided during the installation of the DisDAQ package.

Within the DisDAQ instance on the cluster management software, all necessary apps are installed and configured (e.g., the correct physical ports to which the sensors are connected, need to be defined). Furthermore, the intended data storage solution (local data base or raw binary writers or both) is configured. After installation and configuration, the DisDAQ apps can be started, and the system is ready for operation.

IV. Application Scenarios

Possible application scenarios are manifold since the DisDAQ system is highly modular and adaptable. The main focus of this paper is to show its applicability and advantages for industrial process optimization. For completeness, other reference applications will also be briefly described.

A. Condition Monitoring and Predictive Maintenance

The first application scenario describes the usage of the DisDAQ system for long-term condition monitoring and predictive maintenance. Existing machines are often not sufficiently equipped with sensors to aggregate enough data for development and evaluation of predictive algorithms. Retrofitting is applied to integrate additional sensors into the machines to provide such data. However, the location of the sensors may have a crucial impact on the data quality. Since the optimal location, type, and other important parameters (e.g. sampling rate, resolution, etc.) are often not known in advance, they need to be targeted in feasibility studies and initial measurement campaigns. The DisDAQ system is very well suited for such tasks due to its flexibility and configurability. In the presented application scenario, the overall goal is to determine the wear of a cutting tool from measurement data of the cutting process as described in [9]. In this paper, the authors use different sensors to detect first the operational state of the cutting tool (cutting or not cutting) and to estimate then the wear of the tool. The DisDAQ system is now used gather data from all sensors. Most of them are directly connected to the DisDAQ system (motor current sensor, accelerometers, infrared thermometers) and sampled with a rate of up to 2 kHz. A rotating sensor is integrated via an external interface. Thus, all relevant data is stored together in the DisDAQ system. Using DisDAQ and its feasibility of remotely accessing the acquired data directly since the installation, the overall process (sensor selection, parameterization, algorithm development) is simplified.

B. Process Optimization Scenario

The second application scenario involves the monitoring of a large-scale indoor production line, producing walls of prefabricated houses. Throughout the manufacturing process, wall elements undergo various production steps, in which predominantly manual tasks are performed by one or multiple workers. The complexity of products varies widely, as the prefabricated houses are highly customizable. Consequently, each wall element can be considered as an individual product with customized manufacturing effort. To optimize the overall production efficiency and minimize backlogs at the production line, the allocation of staff and the production sequence of wall elements need to be optimized. To train a digital twin of the plant using actual production data, synchronized data of production times must be recorded. Retrofitting of a distributed recording system using off-the-shelf components is challenging, as wiring of

Figure 4: Sensor setup for roof-top installation of the mobile environmental sensor rack (MESR) using a mobile DisDAQ instance.

components throughout the entire plant is not feasible due to the constant traffic of forklifts, indoor cranes, and customized systems used for transporting and rotating wall elements during production. As solution, several wireless nodes of the DisDAQ system are distributed across the most critical production steps. These nodes record the passage of wall elements by measuring the RSSI (Received Signal Strength Indication) values between RFID readers connected to the DisDAQ nodes and disposable RFID tags mounted onto the wall elements. Customized algorithms process the RSSI data for each station, generating events such as "Wall #a finishes production step #b at time #c". The obtained events are timestamped with a time-base synchronized across all DisDAQ nodes. Since all nodes are wireless, the system can easily be modified or expanded during operation. The recorded data at current is used to simulate and optimize the production planning [10].

C. Mobile Measurement Scenario

The third application scenario uses the DisDAQ system during testing of autonomous driving (AD) functions. AD-functions need to work under harsh weather conditions, e.g., heavy rain or fog. Thus, the AD-functions also have to be tested during these harsh environmental conditions. The actual conditions must be documented during the test drives. In the course of the Test.EPS project [11], the Mobile Environmental Sensor Rack (MESR) has been developed, which can be mounted on top of a vehicle. It operates in parallel to the sensors used for AD and measures the actual weather conditions including temperature, relative humidity, wind speed and wind direction, precipitation intensity and type, road surface condition, and the GNSS-based position. The MESR setup uses an installation of the DisDAQ system. This application scenario highlights the flexibility and configurability of a DisDAQ system which also supports mobile and independent usage.

D. Application Scenario Summary

All three application scenarios are different in the type of sensors the use, different on the requirements on data acquisition, and different in the overall goal which is targeted by each application scenario. Thus, it has been shown that the DisDAQ system is suitable not only for industry process optimization but also for other application areas like mobile data acquisition.

V. Conclusion

This paper has introduced DisDAQ, a distributed data acquisition system designed to enhance the efficiency and flexibility of industrial manufacturing processes. The DisDAQ system, built on the ROS2 framework and employing containerized nodes, provides a scalable and modular solution for seamless data collection and preprocessing. By supporting a variety of sensor types and incorporating edge computing capabilities, DisDAQ addresses the critical need for synchronized and reliable data acquisition. The three application scenarios presented further validate DisDAQ's capability to support diverse use cases, ranging from condition monitoring and predictive maintenance, over production process optimization, to mobile application scenarios.

Future work will focus on two aspects, first, the improvement and extension of the DisDAQ system itself by supporting additional sensors and integrating advanced algorithms, and second, on the performance evaluation of the system itself regarding scalability, throughput, and robustness. By continuing to refine and expand DisDAQ, we aim to contribute to the ongoing efforts to optimize industrial processes, ultimately driving greater efficiency, cost savings, and sustainability in manufacturing.

Acknowledgements

This work has been supported in part by the Austrian Federal Ministry of climate action, environment, mobility, innovation and Techno-logy (BMK) through the program “Mobilität der Zukunft (MdZ) – 16. Ausschreibung Automatisierte Mobilität (EUREKA 2020). The Research is also funded by the Hungarian Federal Ministry of Transport, Innovation and Technology, Hungarian Funding Agency. This work has been supported in part by the COMET-K2 Center of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

References

[1] T. Lins, R. Augusto Rabelo Oliveira, L. H. A. Correia and J. Sa Silva, "Industry 4.0 Retrofitting," *2018 VIII Brazilian Symposium on Computing Systems Engineering*, Salvador, Brazil, 2018, pp. 8-15, doi: 10.1109/SBESC.2018.00011.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724–4734, 2018.

[3] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125–1142, 2017.

[4] A. Berger, T. Hölzl, L. B. Hörmann, H. -P. Bernhard, A. Springer and P. Priller, "An environmentally powered wireless sensor node for high precision temperature measurements," *2017 IEEE Sensors Applications Symposium (SAS)*, Glassboro, NJ, USA, 2017, pp. 1-6, doi: 10.1109/SAS.2017.7894039.

[5] Y. Zhang, C. Wurll and B. Hein, "KubeROS: A Unified Platform for Automated and Scalable Deployment of ROS2-based Multi-Robot Applications," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 9097-9103, doi: 10.1109/ICRA48891.2023.10160632.

[6] J. Ichnowski *et al*., "FogROS2: An Adaptive Platform for Cloud and Fog Robotics Using ROS 2," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 5493-5500, doi: 10.1109/ICRA48891.2023.10161307.

[7] V. Bode, D. Buettner, T. Preclik, C. Trinitis, and M. Schulz, “Systematic Analysis of DDS Implementations,” *In Proceedings of the 24th International Middleware Conference (Middleware '23)*, New York, NY, USA, 234–246, doi: 10.1145/3590140.3629118.

[8] Open Robotics, “Using Fast DDS Discovery Server as discovery protocol [community-contributed],” https://docs.ros.org/en/humble/Tutorials/Advanced/Discovery-Server/Discovery-Server.html#fast-dds-discovery-server-v2, retrieved May 2024.

[9] K. Pichler, M. Huemer, G. Kaineder, R. Schlosser, B. Dorfner, and C. Kastl, “Wear detection for a cutting tool based on feature extraction and multivariate regression”, 2024 IEEE 25th International Conference on Information Reuse and Integration for Data Science (IRI), accepted, to be published.

[10] M. A. Gartner, W. Grenzfurtner, B. Zauner, M. Gronalt, "Job and Product Rotation for Maximising the Production Output on Multi Mixed-Model Assembly Lines for Element Prefabrication in Industrialised Housebuilding". Computers & Industrial Engineering, Volume 190, 2024. https://doi.org/10.1016/j.cie.2024.110041.

[11] J. Reckenzaun et al., "Transnational Testing, Operation and Certification of Automated Driving Systems: Perspective from testEPS and Central System EUREKA Projects - Mid-Term Results," 2023 IEEE International Automated Vehicle Validation Conference (IAVVC), Austin, TX, USA, 2023, pp. 1-8, doi: 10.1109/IAVVC57316.2023.10328093.