Manufacturing Letters 27 (2021) 39–41

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

Edge-cloud computing performance benchmarking for IoT based machinery vibration monitoring

Ankur Verma [a], Ayush Goyal [b], Soundar Kumara [a,*], Thomas Kurfess [c]

[a] Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA 16801, USA
[b] Optum Global Solutions, Hyderabad, India
[c] Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

ARTICLE INFO

Article history:
Received 10 August 2020
Accepted 7 December 2020
Available online 11 December 2020

Keywords:
Edge computing
Machinery vibration monitoring
Smart manufacturing
Internet of Things

ABSTRACT

Advances in low cost and reliable sensing, connectivity (Internet of Things), computational power, and advanced analytics, are leading to a new wave of innovation in machinery status sensing and condition monitoring. Significant research efforts are directed towards cloud computing architectures. However, given the latency, bandwidth, cost, security, and privacy concerns, further supported by the ever-increasing capabilities of edge computing devices, there is a need to consider both edge and cloud computing together to make informed decisions based upon context and performance. We present an edge-cloud performance evaluation for IoT based machinery vibration monitoring, to foster deployment for the contexts considered.

1. Introduction

Increased visibility into industrial machinery and assets offers the first step towards realizing smart manufacturing. Low-cost and reliable sensors, connectivity (Internet of Things), edge and cloud computing, and data analytics enable the leveraging of real-time data for inferring actionable insights to improve performance metrics [1,2]. Weiss et al. have provided a cost benefit analysis of manufacturing machinery maintenance, underscoring the vast potential and opportunity [3]. Helu et al. highlight the opportunities for and barriers to the deployment of digital and Prognostics and Health Management (PHM) technologies for Small and Medium Enterprises (SME). They conclude a strong SME interest towards basic equipment performance and other capabilities like predictive maintenance and dynamic scheduling. They recommend implementation of appropriate use cases to address various research questions [4].

Vibration monitoring is the most important and commonly used method in machinery condition monitoring [5]. Both, signal processing and deep learning-based approaches are available and are being proposed for machinery vibration monitoring and other similar applications. Much of this practice has involved data acquisition and pre-processing on the edge device and analytics on a more powerful remote computing device or cloud [2,6], while some work has been carried out on the edge device itself [7,8].

Considering the low latency requirements, bandwidth availability, cost, security, and privacy concerns, there is a need for deployment of edge platforms that can operate at the source of the data [9]. In an attempt to study commercial edge IoT platforms, Das et al. [10] have compared the performance of Amazon AWS Greengrass and Microsoft Azure IoT edge with their cloud-only implementations. They report that edge computing is a promising alternative to cloud computing for CPU light workloads, like image recognition (using a small size model and low compute footprint classifier) and scalar values (sensor emulator). Further, they observe that both the platforms do not handle very high throughput messaging well yet. There is a need for *vendor-agnostic* performance evaluation of edge IoT hardware and software from first principles, to stimulate rapid innovation in the open-source community for best performance and architecture identification. This will help foster smart manufacturing capabilities for legacy machines and SME, which are important intended beneficiaries of smart manufacturing technologies [11,12]. Leiserson et al. have identified software performance engineering, new algorithms and hardware streamlining as three themes for computing speed-ups post-Moore era. They expect more special purpose devices for different application domains [13]. Concurrently, advanced analytics such as deep learning are gaining momentum to be deployed at the edge. Therefore, it is important to evaluate edge computing performance.

* Corresponding author.
E-mail address: skumara@psu.edu (S. Kumara).

https://doi.org/10.1016/j.mfglet.2020.12.004
2213-8463/

This requires a benchmarking study to (i) identify the metrics for performance evaluation and, (ii) compare edge and cloud platforms.

In this paper, we demonstrate a benchmarking study using an IoT-based machinery vibration monitoring use-case. We use a triaxial MEMS accelerometer (ADXL-345) sampled at 1600 Hz, interfaced with a Raspberry pi 3 [14] (edge device). 100 experimental runs of 10 s each were carried out, generating approximately 350 KB of data in each 10 s sampling interval. The study has been carried out using commercial off-the-shelf hardware and open-source software (Python), which is in compliance with industry 4.0 ideas of transparency, interoperability and scalability. For time performance evaluation of edge and cloud platforms; we calculate the following:

1. Time-domain features

(Mean, maximum, minimum, range, root mean square, kurtosis, skewness).

2. Fast-Fourier Transform

The main contributions of this work are (i) proposing the performance metrics for edge and cloud computing, for machinery condition monitoring (ii) validation of the proposed architecture using a machinery vibration monitoring case study. A schematic of the benchmarking architecture developed in this work is shown in Fig. 1.

2. Computing methodology

2.1. Edge computing

A Raspberry-pi 3 (RPi) (Quad-Core 1.2 GHz Broadcom BCM2837 64bit CPU 1 GB RAM) running Raspbian operating system, with a 16 GB microSD memory card is the edge device used for interfacing the ADXL-345 using i2c protocol [15]. A Python program running in the RPi was used to acquire the data, compute the time-domain features and the Fast-Fourier Transform (FFT). FFT was plotted on a Virtual Network Computing (VNC) device using Python's Matplotlib library.

The following times are considered for edge computing performance evaluation:

(i) Time for loading libraries
(ii) Time overhead for data acquisition
(iii) Time for computing time-domain features
(iv) Time for computing FFT
(v) Time for plotting graph

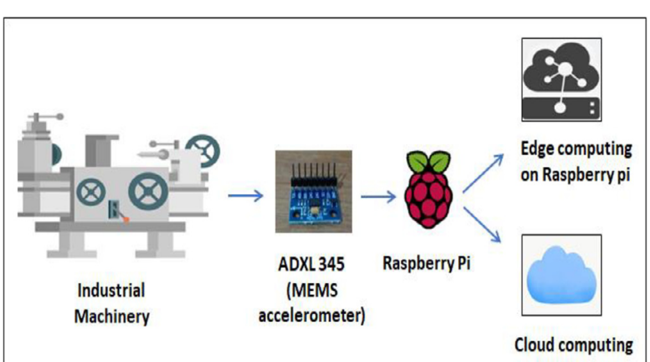

Fig. 1. Edge-cloud benchmarking architecture.

For establishing performance metrics on the edge, time for loading libraries and time for plotting graphs are not included. The rationale for excluding the time for loading task specific libraries is that they are loaded only once when the code is executed. The average time for loading libraries on the RPi is observed to be 1.72 s. Further, the time for plotting graphs is also excluded from performance metrics as it depends on their type (rendering requirements) and quantity. Moreover, graphs may not be always necessary, as fault classification on the edge using deep learning will return only a class label. The average time for plotting FFT plots in this work is 0.54 s. The edge computing performance metrics are shown as a boxplot in Fig. 2. A large variance is found in the computing time for FFT, whereas the computing time for time-domain features is small and consistent in comparison to the former. The variation of the FFT compute time is due to the fact that the FFT algorithm is run on the processor at a lower priority than a number of service routines, and a variable number of service routine interrupts may occur during the FFT calculations.

2.2. Cloud computing

To maintain consistency in experimentation, data acquisition is performed for the same specifications as mentioned in the introduction. Next, the raw data collected in batches of 10 s (approximately 350 KB data per batch) are sent to a PostgreSQL [16] open-source database using wi-fi. These data are then retrieved from the database, the time-domain features and FFT are computed on Heroku [17] cloud platform, running on an Amazon Elastic Compute Cloud (EC2) instance [18]. The entire process is designed accounting for the real-time capabilities required for machinery vibration monitoring and is written in Python, to keep consistency and ease of a single open-source language.

As the raw data leave the edge device, the following times are considered for cloud computing performance evaluation:

(i) Time for insertion into database (T_i)
(ii) Time for retrieval from database (T_r)
(iii) Time for computing time-domain features
(iv) Time for computing FFT

The cloud computing performance metrics are shown as a boxplot in Fig. 3. Insertion time (T_i) includes the data transmission time (from Raspberry pi to the database), and the time to insert into the database. Retrieval time denotes the time taken to retrieve

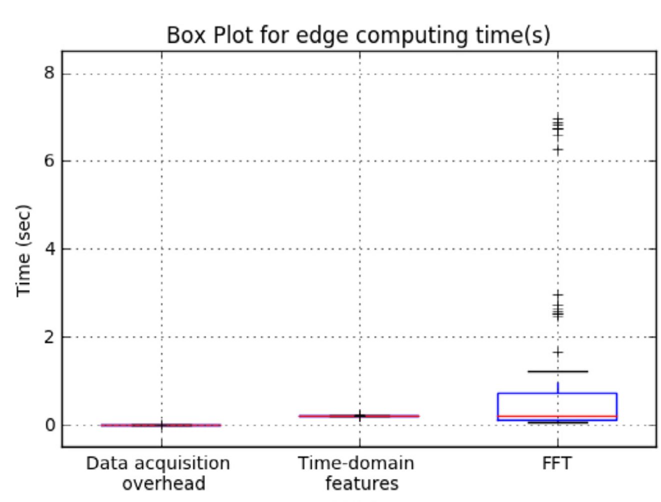

Fig. 2. Edge computing metrics.

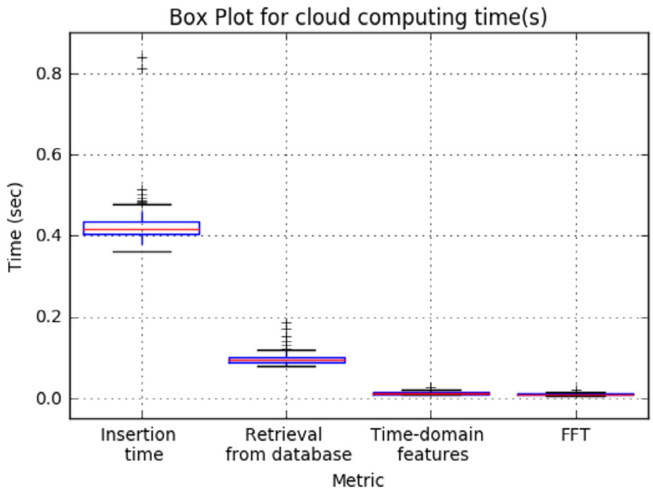

Fig. 3. Cloud computing metrics.

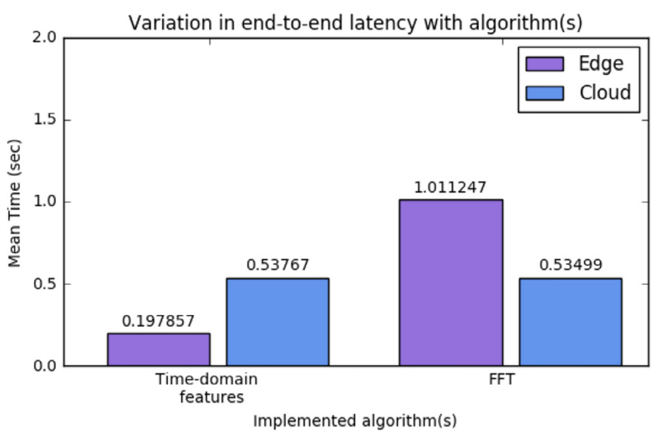

Fig. 4. Variation in end-to-end latency with algorithm(s).

data from the database. Large variance is observed in the database insertion and retrieval times, which may be attributed to transmission and database related reasons. The time for computing FFT and time-domain features was observed to be similar in their central tendencies, unlike that on the edge device. This implies that in our experiments, cloud is better than edge for computing FFT.

3. Performance comparison

The end-to-end latencies for computing time-domain features and FFT, on edge and cloud platforms are shown in Fig. 4.

In our experiments, it is observed that the end-to-end latency for computing time-domain features is less on the edge in comparison to cloud, while this trend reverses in the case of FFT. As the edge computation time (T_a(edge)) increases, the overheads associated with cloud ($T_i + T_r$) diminish in comparison to the increase in edge computation time, making cloud the favorable choice. Considering only end-to-end latency as the metric, edge computing can be preferred over cloud if:

$$T_a(\mathrm{edge}) < T_i + T_r + T_a(\mathrm{cloud}), \quad (1)$$

where T_a (device) = computation time of an algorithm on that device.

4. Conclusions

The benchmarking of edge and cloud computing latencies for an IoT-based machinery vibration monitoring case study was performed using commercial-off-the-shelf hardware and open-source software. The study shows that the choice of edge or cloud platform depends on the computation time of an algorithm and overheads, considering end-to-end latency as the metric. Furthermore, this study identifies the edge and cloud performance metrics from first principles, which can be used for other similar applications in smart manufacturing. There is a need to test the time performance of different existing and new algorithms on various edge devices, given the proliferation of deep learning frameworks for edge analytics. Comparing different edge device performances with the proposed platform and establishing best architectures for different use-cases will enable rapid innovation in the development and deployment of open platforms.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Yang H, Kumara S, Bukkapatnam STS, Tsung F. The internet of things for smart manufacturing: a review. IISE Trans 2019;51(11):1190–216.
[2] Narayanan A, Kanyuck A, Gupta SK, Rachuri S, Machine condition detection for milling operations using low cost ambient sensors. In: Proceedings of the ASME 2016 11th International Manufacturing Science and Engineering Conference. Volume 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing. Blacksburg, Virginia, USA.
[3] Weiss B, Thomas D. Economics of Manufacturing Machinery Maintenance A Survey and Analysis of U.S. Costs and Benefits, National Institute of Standards and Technology Advanced Manufacturing Series 100-34 Natl. Inst. Stand. Technol. Adv. Man. Ser. June 2020, 100-34, 52 pages.
[4] Helu M, Weiss B. The Current State of Sensing, Health Management, and Control for Small-To-Medium-Sized Manufacturers. In: Proc ASME Int Conf Manuf Sci Eng.; 2016, 2:MSEC2016-8783
[5] Mohanty AR. Machinery Condition Monitoring: Principles and Practices. CRC Press, Taylor and Francis Group; 2014.
[6] Koene I, Viitala R, Kuosmanen P. Internet of things based monitoring of large rotor vibration with a microelectromechanical systems accelerometer. IEEE Access 2019;7:92210–9.
[7] Lynn R, Wescoat E, Han D, Kurfess T, Embedded fog computing for high-frequency MTConnect data analytics, Manuf Lett, 2018;15(Part B):135–8.
[8] Nguyen V, Dugenske A, An I2C based architecture for monitoring legacy manufacturing equipment, Manufacturing Letters, 2018;15(Part B):67–70.
[9] Georgakopoulos D, Jayaraman PP, Fazia M, Villari M, Ranjan R. Internet of things and edge cloud computing roadmap for manufacturing. IEEE Cloud Comput 2016;3(4):66–73.
[10] Das A, Patterson S, Wittie M, EdgeBench: Benchmarking Edge Computing Platforms. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich; 2018, p. 175–80.
[11] Strategy for American Leadership in Advanced Manufacturing, National Science and Technology Council, 2018.
[12] Fourth Industrial Revolution: Beacons of Technology and Innovation in Manufacturing, World Economic Forum in collaboration with McKinsey and Company; 2019.
[13] Leiserson C, Thompson N, Emer J, Kuszmaul B, Lampson B, Sanchez D, et al. There's plenty of room at the Top: What will drive computer performance after Moore's law?. Science 2020;368(6495).
[14] Raspberry Pi 3, raspberrypi.org, 2020
[15] Adafruit ADXL-345 library: https://github.com/adafruit/Adafruit_Python_ADXL345/ [accessed on 9 July 2020].
[16] PostgreSQL, postgresql.org; 2020.
[17] Heroku, heroku.com; 2020.
[18] Amazon EC2, aws.amazon.com; 2020.