



13th Conference on Cloud Computing, Big Data & Emerging Topics (JCC-BD&ET 2025)

[HTTPS://JCC.INFO.UNLP.EDU.AR](https://jcc.info.unlp.edu.ar)



@CONF_CC_BD_ET



JCC@LIDI.INFO.UNLP.EDU.AR



POSTGRADO

FACULTAD DE INFORMÁTICA



III-LIDI
INSTITUTO DE INVESTIGACIÓN
EN INFORMÁTICA - LIDI

13th Conference on Cloud Computing Conference, Big Data & Emerging Topics (JCC-BD&ET 2025)

La Plata, Buenos Aires, Argentina.
June 24–26, 2025

13th Conference on Cloud Computing, Big Data & Emerging Topics (JCC-BD&ET 2025) ; Compilación de Marcelo Ricardo Naiouf ... [et al.]. - 1a ed. - La Plata : Universidad Nacional de La Plata, 2025.

Libro digital, PDF

Archivo Digital: descarga y online
ISBN 978-950-34-2583-1

ISBN 978-950-34-2583-1



1. Actas de Congresos. 2. Computación. 3. Conferencias. I. Naiouf, Marcelo Ricardo, comp.
CDD 004.01

Table of Contents

Full papers	1
Development and Implementation of an AWS-Based Platform for Automated Prediction of Antibiotic Resistance from Mass Spectrometry Profiles.	2
<i>M. Flores Estay (UCM – Chile), X. A. López Cortés (UCM – Chile), J. I. Bernal Osses (UTALCA – Chile), J. M. Manríquez-Troncoso (UCM – Chile).</i>	
Orthogonal Moments-Based Feature Extraction for MRI Classification.	21
<i>G. Degiuseppe (UCA – Argentina), A. Quintero-Rincón (UCA – Argentina).</i>	
Pipeline to detect spike-and-wave EEG patterns based on polynomial regression modeling and Taylor series feature selection.	35
<i>M. F. Adell (UCA – Argentina), J. Balda (UCA – Argentina), F. Casas (UCA – Argentina), C D’Giano (FLENI – Argentina), A. Quintero-Rincón (UCA / FLENI – Argentina).</i>	
Rule-Based Matching for Real Estate Features Detection.	49
<i>M. A. Ibañez Gutkin (UNLP – Argentina), A. A. Pagano (UNLP – Argentina), L. Tanevitch (UNLP – Argentina), D. Torres (UNLP / UNQ – Argentina).</i>	
WEEE Prediction Model Based on Neural Networks.	63
<i>J. Facuy (UAE – Ecuador), A. Pasini (UNLP – Argentina), E. Estévez (UNS / UNLP – Argentina), C. Moran (UAE – Ecuador).</i>	
Short papers	74
Automated and Secure Login in ALERTAR, a Resilient Cloud–Fog–Edge mHealth System for Hospital Environments.	75
<i>C. Zanellato (UNCOMA – Argentina), R. Cañibano (UNCOMA – Argentina), J. Balladini (UNCOMA – Argentina)</i>	
Contributions to the modeling and simulation of an automatic package classification system to improve decision-making in line balancing.	80
<i>E. Acosta (UNGS – Argentina), J. A. de Queiroz (UNIFEI – Brasil), A. Gaudiani (UNGS – Argentina).</i>	
Development of a Hand Motion Sensing Glove for Exergames: Design Evolution and Future Perspectives.	85
<i>A. Del Gener (UNLP – Argentina), C. Sanz (UNLP – Argentina).</i>	

Ensuring Quality in the OECD AI Lifecycle Through ISO/IEC Standards.	90
<i>J. I. Torres (UNLP – Argentina), A. Pasini (UNLP – Argentina), P. Pesado (UNLP – Argentina).</i>	
Internet deception to share IoC.	96
<i>P. G. Maddalena Kreff (UNLP – Argentina), P. Venosa (UNLP – Argentina), P. Bazán (UNLP – Argentina).</i>	
Interpretable Machine Learning for Real Estate Valuation: A Case Study with Small Data.	101
<i>E. Gutiérrez (UNS – Argentina), L. C. López del Río (US – España), J. M. C. Ocerin (iManagement & Tourism – España).</i>	
No Latency, No Waste: How Fog Computing Optimizes Precision Agriculture.	106
<i>R. Gonçalves (UAI – Argentina), G. Rossi (UAI / UNLP – Argentina).</i>	

No Latency, No Waste: How Fog Computing Optimizes Precision Agriculture

Gonçalves, Ricardo¹[0009-0005-8346-4808] Rossi, Gustavo^{1,2}[0000-0002-3348-2144]

¹ Universidad Abierta Interamericana. Facultad de Tecnología Informática.
Centro de Altos Estudios en Tecnología Informática - CAETI

² LIFIA. Facultad de Informática, UNLP
ricardo.goncalves@alumnos.uai.edu.ar,
gustavo@lifia.info.unlp.edu.ar

Abstract. The implementation of Internet of Things (IoT) technologies in precision agriculture has been revolutionizing crop resource management. Soil sensors continuously monitor moisture, temperature, and salinity, providing crucial data for precise irrigation and soil management, while weather sensors track atmospheric conditions including air temperature, humidity, precipitation, and wind speed. This data enables prediction and management of pests and diseases, while also informing harvest decisions. Continuous monitoring and advanced data analysis allow for identification of trends and anomalies, facilitating rapid and precise adjustments to agricultural operations. In this article we propose an approach to deal with the problem of latency in the use of IoT in remote areas.

Keywords: IoT, Latency, Fog Computing, Grafana, Monitoring, Precision Farming.

1 Introduction

Precision Agriculture, also known as Precision Farming, is an innovative and detailed approach to agricultural management that uses advanced technologies to optimize production. This methodology employs sensors, satellite imagery, drones, and GPS systems to collect field-specific data[1]. Fog Computing (also called edge cloud computing) is a service distribution paradigm for edge computing systems. It creates an intermediate layer between cloud infrastructure and network edge devices, positioned closer to endpoint equipment. This architecture plays a critical role in supporting real-time application execution and low-latency processing requirements [2]. In both scenarios, latency emerges as a critical factor. In IoT environments, latency plays an even more significant role due to the inherent nature of these systems, which often involve resource-constrained devices, wireless communication, and time-sensitive applications. This study presents an experimental framework to evaluate latency in cloud versus fog computing architectures. The paper is organized into five sessions: Session 2 establishes the motivation, Session 3 details the case study, Session 4 describes the testing methodology, and Session 5 presents the obtained results.

2 Our Approach

This work forms part of the master research carried out by Ricardo Gonçalves at UAI; we seek to address the critical challenge of latency in networks and IoT systems in precision agriculture through a proposal for the use and implementation of networks in Fog Computing. Based on architectural principles established by Bonomi et al. (2012) and later milestones of the OpenFog Consortium, the study presents an empirical evaluation that compares paradigms for the use of computing in the cloud and in the cloud in the agricultural context. To validate the research methodology, we created an experimental implementation based on AWS, where we sought to replicate one of the main operational restrictions of rural environments, which is geographic dispersion, obtained through the positioning of multi-regional AWS nodes, including humidity, temperature and lighting sensors in it. Our results demonstrated conclusive evidence that Fog nodes achieve a 70–85% reduction in end-to-end latency compared to traditional cloud processing, measured using RTT (57–140 ms vs. 300–400 ms) and one-way (3–4 ms vs. 220–225 ms) metrics. These findings not only validate the technical feasibility of Fog Computing for time-sensitive agricultural operations in resource-constrained environments, but also establish a new benchmarking methodology for evaluating edge computing performance in agricultural applications, particularly relevant for regions where reliable cloud connectivity remains an issue.

3 Case Study

We seek to use a testing environment closer to physical reality, with sensors and virtual servers. To implement the experiment, we created three servers with different roles within the cloud and fog computing architecture. The experimental setup involved deploying the following servers and devices within the Amazon AWS environment, as described below:

Server Configuration:

- **Raspberry Server (IoT Gateway):** This server acts as an entry point for sensor data, being responsible for connecting to AWS IoT Core and sending information to Fog and Cloud servers;
- **Fog Server:** Implemented as an instance on AWS, this server processes data locally before sending it to the cloud, which reduces latency and overhead on the cloud infrastructure;

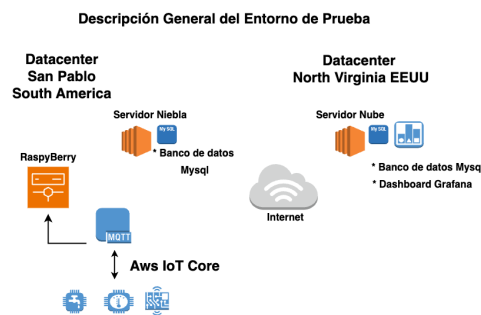
- **Cloud Server:** Instantiate on AWS with a centralized database, where collected and processed data is stored for subsequent analysis;

Aws IoT Core Virtual Sensors:

We create virtual sensors in AWS IoT Core to simulate physical sensors in a real environment. These sensors generate temperature, humidity and luminosity data, which are sent to RaspBerry via the MQTT protocol. The configuration consists of:

- Configuration of MQTT “topics” for each sensor;
- Creation of digital certificate for devices;
- Configuration of security policies to allow communication between devices and AWS;
- Configuration of an MQTT client on RaspBerry to receive and resend data.

Test environment:



4 Test Development

We developed six scripts, all storing data in a database. This storage enables data comparison and report generation, with results being sent to Grafana. All scripts can read sensors in AWS IoT Core: (Sensor_Humidity, Sensor_Temperature, Sensor_Light). Some scripts use Sensor_Temperature while others use Sensor_Humidity, depending on requirements. The scripts take measurements every 20 seconds and record data in the databases, and this interval is established to balance continuously updated measurements without overloading the storage and processing systems and being able to generate graphs that can be analyzed. Communication between the virtual sensors and the RaspBerry server is established via the MQTT protocol, using AWS IoT Core as the message broker. To ensure secure data transmission, a set of cryptographic keys was implemented and stored in the RaspBerry server's repository. The system's operational workflow follows these steps: the virtual sensors configured in AWS IoT Core connect and publish to the MQTT server (RaspBerry), which periodically generates measurements of variables including temperature, humidity, and luminosity. The data consists

of randomized values with variable parameters. The Raspberry device, functioning as an intermediate node, receives these messages via MQTT and immediately appends a transmission timestamp, precisely recording when the data was processed. After establishing the connection with the sensors, the script running on the Raspberry Pi generates randomized simulated measurements of temperature, humidity, and light. These data points are then written directly into the MySQL databases on each server, accompanied by their respective timestamps, ensuring a detailed and accurate reading history. The Fog server, physically closer to the Raspberry Pi, receives the data first, enabling faster processing and reducing latency associated with transmission. Meanwhile, the Cloud server—located remotely—also receives the measurements but with inherently longer transmission times due to geographical distance and network characteristics.

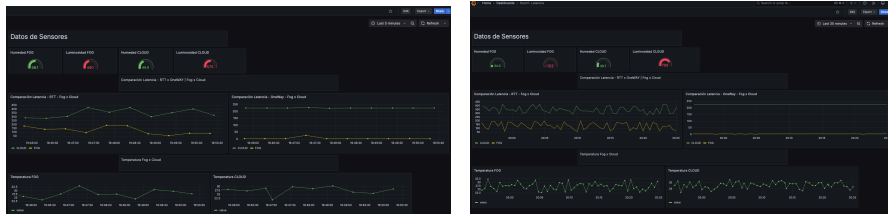
5 Results:

Fog Computing latency was significantly lower than Cloud latency. We tested two models (RTT and ONEWAY). Fog processing occurs closer to the devices, which significantly reduces transmission and storage latency. Results show the average time between a packet's departure from a sensor and its return plus subsequent database registration in the Fog environment ranged between 57 ms and 140 ms using RTT measurement. In contrast, the Cloud environment showed an average latency between 300 ms and 400 ms - considerably higher than Fog Computing using the same methodology. In OneWay packet transmission tests, the average time between departure and database registration for Fog processing ranged between 3-4 ms, while Cloud processing ranged between 220-225 ms.

Result Panel:

To present the collected sample data and carry out the necessary comparisons, we installed Grafana, which has a graphical interface with time stamps and temporal graphs. Grafana uses a graphical interface on a WEB server and connects to databases from both the cloud server and the fog server.

5 and 30 Minutes Menasurement.



6 Related Work:

In [3] The author presents cloud computing as an extension of cloud computing, highlighting its distinctive characteristics such as low latency, on the other hand [4] demonstrates the importance of taking the process to the limit in critical services. In [5] the author makes a comparison between clouds and clouds and use a tool to measure latency. In almost all studies, the topic of latency is of utmost importance. Its importance is clear in environments with a shortage of Internet links or, on the other hand, in environments that require a quick or immediate response. Reducing latency is crucial to success. By bringing processing to the edge of the network, processing it and then sending it to the cloud, we significantly help reduce network usage.

7 Conclusion:

In this article, we seek to present the results of results of experimental tests demonstrating that the use of Fog Computing significantly reduces latency in comparison with an approach based exclusively on Cloud Computing, thereby validating our initial hypothesis. This difference is calculated using key metrics, such as round-trip time (RTT) and response time with methodology (ONEWAY), which confirms that local processing in the cloud cover minimizes delays in transmission and data storage.

References

- [1] E. A. Q. Montoya, S. F. J. Colorado, W. Y. C. Muñoz, and G. E. C. Golondrino, "Propuesta de una arquitectura para agricultura de precisión soportada en IoT," *Revista Ibérica de Sistemas e Tecnologías de Informação*, no. 24, pp. 39–56, 2017.
- [2] A. V Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," *Computer (Long Beach Calif)*, vol. 49, no. 8, pp. 112–116, 2016, doi: 10.1109/MC.2016.245.
- [3] M. A. Orozco, "Análisis de factibilidad en la implementación de un sistema de control para el monitoreo del proceso de fermentación del vino (Doctoral dissertation, SIPI).," 2020. Accessed: May 31, 2023. [Online]. Available: https://repositorio.cetys.mx/bitstream/60000/1102/1/Orozco%20Armando_Proyecto%20final.pdf
- [4] C. V Raghavendran, A. Patil, G. N. Satish, M. Shanmukhi, and B. Madhuravani, "Challenges and opportunities in extending cloud with fog computing," *Int J Eng Technol*, vol. 7, no. 439, pp. 142–146, 2018.
- [5] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 1, pp. 1–21, 2018.