STREAM PROCESSING IN DECENTRALIZED ARCHITECTURES: CHALLENGES AND ADAPTIVE SOLUTIONS ACROSS CLOUD, FOG, AND EDGE

Alireza Faghihi Moghaddam

Department of Information Technology, Uppsala University, Sweden,

ABSTRACT

In recent years, the rapid development of data-driven applications has posed significant challenges for data computation in different domains. Handling and processing continuous data streams have become essential for building data-driven organizations, which places a high burden on traditional computing. As a centralized method, cloud computing often struggles with application latency, mainly because of geographic distance and network bandwidth challenges . The increasing scale and complexity of data, characterized by high volume, velocity, and variety, demand computational infrastructures that are powerful, adaptive, and efficient in terms of pro- cessing. Fog and Edge computing are two decentralized network solutions that move the computation closer to the data source, lowering network traffic while improving the response time. Edge computing performs computations within IoT devices, resulting in real-time data processing and subsequently transferring less time-critical data to the cloud. In contrast, Fog Computing utilizes fog nodes with high computational power for data processing and storage. These nodes are within the same local network, and a decentralized solution is a better choice when working with a large number of IoT devices, the need for local computational power, and storage. Both fog and edge computing rely on cloud infrastructure for long-term data storage and larger computations. This study provides a comprehensive comparative analysis of the Fog, Edge, and Cloud computing paradigms, with a particular focus on their applicability to real-time data stream processing. to determine their strengths and ideal use cases in a table and to showcase their advantages and disadvantages in stream processing

.***KEYWORDS***

Stream processing, Edge computing, Fog computing, Cloud computing.

1. INTRODUCTION

The emergence of data-intensive applications such as autonomous systems, real- time analytics, and Internet of Things (IoT)-based services has made data-stream processing an essential requirement in modern computing. Although batch pro- cessing involves analysing stored data over time, it naturally lacks the real-time decision-making and timely insights required for latency-sensitive applications. In contrast, stream processing involves the ingestion, processing, and analysis of data in motion, introducing new challenges related to the increasing scale, velocity, and distribution of data streams.

David C. Wyld et al. (Eds): CRYPIS, CBIoT, CAIML, NLCA, NC, WiMo, ICAIT, ICDIPV, ITCSE – 2025
pp. 87-95, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151706

Meeting these demands using traditional cloud solutions often leads to higher operational expenditures (OPEX), particularly when dealing with real-time data pro- cessing. The main reasons these organizations need to address their challenges are data transfer, storage, and computing resource usage[1]. Moreover, many applications within the Internet of Things (IoT), 5G networks, and interactive games are now focusing more on their quality to ensure customer satisfaction, which means that they will depend more on low-latency features.

Moreover, the number of real-time IoT applications has been significantly increased. These applications require resources that support fast processing and low access latency to minimize the total response time. Some examples of these applications are autonomous robots and disaster management applications (e.g., natural Hazard Management).

2. BACKGROUND

The following formatting rules must be followed strictly. This (.doc) document may be used as a template for papers prepared using Microsoft Word. Papers not conforming to these requirements may not be published in the conference proceedings.

Cloud computing refers to a scalable, on-demand solution for remote computation over the Internet that uses datacentres. This model provides centralized resource management and orchestration, offering scalable computing capabilities that can expand significantly based on demand, and is best suited for storing large data and computation tasks without requiring time-critical workloads [2]. However, its limitations are due to its reliance on wide-area networks, which often result in high latency and bandwidth consumption. Additionally, concerns regarding data privacy and regulatory compliance have emerged, particularly when data storage and processing cross international boundaries [3].

Cloud computing refers to a scalable, on-demand solution for remote computation over the Internet that uses a datacentre. This model provides centralized resource management and orchestration, offering scalable computing capabilities that can expand significantly based on demand, and is best suited for storing large data and computation tasks without requiring time-critical workloads [2]. However, its limitations are due to its reliance on wide-area networks, which often result in high latency and bandwidth consumption. Additionally, concerns regarding data privacy and regulatory compliance have emerged, particularly when data storage and processing cross international boundaries [3].

In contrast, edge computing moves the computation location near the source, such as sensors, mobile devices, and gateways (i.e., edge nodes). This model reduces the problem of long-distance data transmission and allows real-time processing with low latencies. Unlike centralized cloud computing, edge computing leverages local computation across distributed nodes, even with intermittent or no connectivity to the Internet or cloud[4]. This enhances resilience in remote or unreliable network environments and improves the reliability and trustworthiness of the stream-processing tasks. However, Edge nodes are often resource-constrained, and orchestrating workloads across a heterogeneous network can introduce maintenance challenges[5,6].

Fog computing lies between clouds and edges as an intermediate layer that extends cloud capabilities closer to the data to distribute computational resources across ”fog nodes” near the edge of the network. The key features that distinguish Fog include context awareness, lower latency, mobility support, real-time computation, and heterogeneity[7]. The computational and storage capabilities of fog nodes are often greater than those of edge devices, which results in reduced latency, preprocessing, and filtering in IoT applications that require a fast response[8–10].

In this survey, we built on existing studies to provide a comprehensive overview of the challenges associated with data-stream processing in cloud, fog, and edge computing environments. We examined the architectural differences, processing capabilities, and inherent limitations of each layer to elucidate their impact on the orchestration of the stream processing systems. We categorize the challenges primarily across the resource management, latency, security, and scalability landscapes.

3. CHALLENGES OF DATA STREAM PROCESSING

Data stream processing proposes a paradigm shift from traditional batch processing, driven by the increasing need for real-time insights and responsive decision-making in fast-paced environments. Stream processing plays a vital role when organizations need to obtain useful knowledge from ongoing events efficiently to take appropriate action. However, this process faces significant challenges, which arises from the nature of big data streams that must be addressed. The most critical challenges include:

3.1. Load Balancing

A stream processing system must exhibit self-adaptive behaviour when receiving data faster, slower, or in larger amounts than usual, thereby minimizing the risk of data loss during peak loads[11]. One of the core challenges in building such systems cost-effectively lies in the system’s inability to dedicate sufficient resources to handle peak demands at all times, as such provisioning would lead to the waste of resources during normal load periods. Using a distributed computing paradigm, such as Fog or Edge computing, allows the offloading of part of the data to a datacentre whenever the local system becomes overloaded [11].

3.2. Storage and Ingestion

The separation of data ingestion, processing, and storage layers across heterogeneous systems has been widely recognized as a key performance bottleneck in distributed data stream processing, especially within fog and edge computing environments, where high-velocity data are continuously produced by IoT devices [12]. Disjointed architectures often result in redundant I/O operations, higher latency, and more complex resource management owing to movement between different systems. This poses a significant challenge for edge computing, where data arrive constantly and unpredictably from many IoT devices[13]. By combining these separate components into one integrated system, the overall architecture becomes simpler, and data can flow through the system more easily.

3.3. Privacy

Stream processing systems face significant privacy challenges owing to their Realtime data handling capabilities, particularly when processing sensitive information across multilayered architectures. Sensitive data can be intercepted at various layers (sensor, fog, and cloud), leading to potential breaches or unauthorized access[14]. It is more feasible to deploy robust and comprehensive security protocols in the cloud, where powerful hardware can support advanced authentication, encryption, and monitoring techniques. In contrast, edge and fog computing systems are often under strict resource constraints, making them more vulnerable to breaches.

4. Comparing Fog, Edge, and Cloud Computing Environments

In this chapter, we compare Fog, Edge, and Cloud computing environments, discussing the details of their characteristics, which define how each computing strategy stands out and is best suited for specific scenarios. The results are presented in Table 1.

Table 1. Mapping of Proposed Solutions to Computing Paradigms.

Characteristic	**Fog Computing**	**Edge Computing**	**Cloud Computing**
Computation Location	Fog nodes within the local area network	Inside the IoT Devices	Cloud datacentres
Latency	Low	Instant (within IoT)	High
Real-Time Support	Supported	Supported	Limited
Response Rate	Milliseconds, sub seconds	Milliseconds	up to Minutes
Storage Capacity	High	Very Low	Unlimited(datacentre level)
Computational Power	High	Limited	Unlimited(datacentre level)
Geographical Location	Local Network	Device Level	Datacentre
Scalability	Moderate	Limited	High
Fault Tolerance	High	High	Moderate
Solution Distribution	Decentralized over Local Network	Decentralized at IoT Device Level	Centralized at Datacentres
Energy Efficiency	High	High	Low
Cybersecurity	Decentralized solution with heterogeneous nature, potentially leading to increased attack surface, low authentication, and limited visibility in the network	Most decentralized with only one device's data, limited cybersecurity features	Datacentre security level, centralized solution with all data at one location, more flexibility for developing security protocols
Characteristic	**Fog Computing**	**Edge Computing**	**Cloud Computing**
Use Case	Local	Local	Global
Distance to End-nodes	Close	Very Close	Far
Location Awareness	Supported	Supported	Not Supported
Heterogeneity Support	Supported	Supported	Supported
Maintenance Complexity	High	Moderate	Low

General Application	Where the number of IoT devices is high and local/offline computation power and storage are required; e.g., Smart factories, real-time healthcare, autonomous vehicles, Smart grids	Where the number of IoT devices is small/limited and real-time response is demanded; e.g., special use in Smart factories, healthcare, traffic control	Ability to compute and store a high amount of data and provide different service platforms (IaaS, PaaS, SaaS)

4.1. Computation Location and Latency Challenges

In data stream processing for Edge, Fog, and Cloud environments, the computation location defines where the data are processed. Edge and Fog computing prioritize processing near the data source to minimize latency, whereas cloud computing transmits data to remote datacentres.

Edge computing handles real-time processing inside IoT devices, which are the primary source of data, enabling real-time processing but with limited computing power and energy resources[9]. Fog computing introduces a different hardware solution, fog nodes(such as fog servers and gateways), with greater computational power to support tasks such as aggregation and preliminary analytics in local networks [15]. This model is particularly effective in dense IoT deployments, enabling local storage and reducing the dependency on distant servers [16,9].

Nevertheless, both Edge and Fog often offload data to cloud infrastructure for advanced analysis and long-term storage. Although cloud computing offers virtually unlimited computational resources and storage capacity, it incurs increased latency and potential security risks owing to data transmission over wide area networks [17].

4.2. Data Volume and Scalability

Managing the massive and continuous influx of data generated by IoT devices, sensors, and user applications is a primary challenge in data-stream processing across different computing environments. Limited computational resources, memory capacity, and energy constraints have made it difficult to efficiently handle largescale, high-velocity data streams [13]. Although Fog nodes provide more resources than edge devices, they are still significantly constrained compared to centralized cloud infrastructures[18].

Using distributed storage systems and elastic resource provisioning, cloud computing can provide nearly unlimited scalability, thereby allowing the effective processing and analysis of large data streams [19]. However, businesses cannot rely solely on the cloud because of latency and bandwidth challenges when transmitting vast volumes of raw data from distributed sources.

4.3. Resource Constraints and Reliability

Resource limitations are a fundamental challenge in data stream processing in edgeand fog-computing environments. These devices often offer limited CPU capacity, storage, and energy reserves, restricting their ability to continuously process highvolume data streams.

Unlike centralized cloud environments, where redundancy and fault tolerance mechanisms are mature, Edge and Fog layers require lightweight, decentralized approaches to ensure system reliability without overburdening limited resources [22].

Autoscaling mechanisms have been widely adopted to address dynamic workload fluctuations in cloud computing systems. However, traditional autoscaling methods, such as the horizontal pod autoscaler (HPA), rely on reactive scaling based on hardware utilization metrics(e.g., CPU or memory usage), which may not be sufficient for heterogeneous and resource-constrained edge computing environments. To address these limitations, a recent study by Ju et al.[23] proposed a solution that uses a prediction method for the workload using information on system resource usage (e.g., CPU, RAM, and I/O) and demands in advance. In this way, we can maintain the system performance without introducing heavy overhead, making it particularly suited for the reliability and efficiency challenges faced in Edge and Fog computing systems.

5. POTENTIAL SOLUTIONS AND PROPOSED SYSTEM

In this section, we synthesize existing adaptive mechanisms from recent literature, focusing on their applicability to stream processing in edge, fog, and cloud environments. While we do not propose new algorithms, we compare these approaches to identify gaps and practical considerations for deployment.

5.1. Reactive and Proactive Autoscalers

By employing data on the workload of computing systems, such as CPU utilization, request rates, and memory usage, autoscalers can dynamically adjust the number of running instances to meet performance requirements[24]. Reactive autoscalers respond to changes in metrics after they occur; for instance, they scale out when the CPU usage exceeds a threshold with the HPA. This approach is simple to implement but may lead to delayed reactions and temporary performance degradation during sudden demand spikes [25].

However, relying solely on reactive methods may be insufficient for systems that require consistent performance under highly dynamic workloads. To address this limitation, proactive autoscalers attempt to anticipate future resource demands based on historical data, workload trends, or external inputs, such as time-of-day patterns or user behaviour forecasts.

One such approach is the Proactive Pod Autoscaler (PPA), which integrates a machine-learning-based forecasting mechanism into the Kubernetes autoscaling loop[23]. The PPA maintains historical metric logs and a continuously updated prediction model to estimate the optimal number of pod replicas in advance, thereby ensuring data processing quality.

5.2. Task Scheduling

Task scheduling plays a pivotal role in ensuring efficient and timely processing of data streams in distributed computing environments spanning edge, fog, and cloud layers. The heterogeneous nature of devices in the edge layer results in variations in resource capacities and dynamic workload patterns. Determining when, where, and how tasks should be executed is a nontrivial challenge that requires further research. In task scheduling, we aim to optimize performance metrics such as latency, energy, and cost. For instance, in a complex and constantly changing environment similar to smart manufacturing, heuristic algorithms are used to map tasks to fog nodes with minimal time and energy costs [26]. These algorithms aim to minimize the makespan and energy cost by considering parameters such as task size, execution time, node availability, and current system load[27]. Other methods, such as that proposed in[28], employ reinforcement learning to manage the complexity of realtime task offloading in dynamic edge computing environments. By tracking changes in the network and task requirements, the system decides to send its resources. This helps keep delays low and makes better use of the available resources so that more tasks can be handled successfully and rapidly.

5.3. Lightweight Privacy Protocols

Employing security protocols on IoT devices can help prevent many disastrous attacks, including Denial of Service (DoS) attacks, session hijacking, Eavesdropping, or Data tampering attacks[29]. One of the challenges faced by decentralized computing methods is data privacy, which is due to resource-constrained devices at the edge layer; therefore, implementing the same protocol on cloud computing is not feasible for edge and fog. In such environments, lightweight and adaptive security mechanisms are required to balance the trade-off between computational overhead and protection requirements. For instance, Datagram Transport Layer Security (DTLS) is often preferred over traditional TLS in edge computing because it is an extension of the User Datagram Protocol (UDP)[30]. One such protocol is the Constrained Application Protocol (CoAP), which is designed for resource constrained devices and networks. CoAP is a client/server model that enables the client to request services from the server as needed, and the server responds to the client's request without acknowledgments for every message, which helps preserve energy in resource-constrained devices. . However, DTLS can also pose challenges, such as increased latency and handshake complexity. Therefore, the design of secure edge systems often involves customized or tiered approaches to authentication, encryption, and access control, which are tailored to device capabilities and context specific risk levels.

Table 2: Mapping of Proposed Solutions to Computing Paradigms

Computing Layer	Suitable Solutions	Notes
Cloud	– Reactive Autoscalers (e.g., HPA) – Proactive Autoscalers (e.g., PPA with ML-based forecasting)	Abundant resources allow for sophisticated scaling strategies, including predictive models and workload forecasting.
Edge	– Reinforcement Learning-based Task Scheduling – Lightweight Privacy Protocols (e.g., DTLS, CoAP)	Limited computational resources require efficient adaptive scheduling and minimal-overhead privacy mechanisms.
Computing Layer	**Suitable Solutions**	**Notes**
Fog	– Heuristic Task Scheduling (e.g., makespan and energy-aware) – Tiered Lightweight Privacy Mechanisms	Moderate resources and proximity to both the cloud and edge make them suitable for hybrid approaches.

6. FUTURE DIRECTIONS

The increasing demand for real-time, distributed stream processing across Cloud, Fog, and Edge computing layers opens numerous research directions, some of which are as follows:

- **Privacy-Preserving Stream Analytics:** Lightweight privacy protocols like DTLS and CoAP can serve as potential solutions in addressing challenges with implementing heavy security protocols on IoT devices , but there's a need for customizable privacy-preserving analytics pipelines that adapt encryption levels based on context, device capabilities, and data sensitivity without compromising performance.

- **Federated Learning for Stream Adaptation:** Applying federated learning to real-time stream data could enable edge nodes to learn local patterns and contribute to global models without transmitting raw data, balancing privacy and performance.
- **Benchmarking Standards for Heterogeneous Environments:** There is a lack of standard benchmarking tools that can accurately measure stream processing performance across heterogeneous cloud-fog-edge stacks. Future studies should use simulation environments to evaluate latency, fault tolerance, and cost-effectiveness.

7. CONCLUSION

Cloud computing has been a significant achievement in the IT industry because it provides high storage capacity and computational power on demand. However, the need for real-time computation and decision-making on remote devices has resulted in new problems, especially as smart societies encounter issues such as rapid response requirements, bandwidth limitations, latency, and real-time processing constraints. Consequently, fields with requirements similar to those of stream processing require solutions to meet their demands. In response, network paradigms are changing from centralized to decentralized approaches to decrease network traffic and improve robustness. Decentralized methods, such as fog and edge computing, can result in faster response rates and lower network traffic levels. This study examined the underlying reasons and challenges associated with stream processing across three computational layers of a decentralized architecture: cloud, fog, and edge computing. It provides a comparative analysis of these models and proposes potential solutions to the identified challenges, drawing on existing research that has addressed similar issues and demonstrated improvements in shared problem areas.

REFERENCES

[1] J. Zhang, L. Cheng, C. Liu, Z. Zhao, and Y. Mao, "Cost-aware scheduling systems for realtime workflows in cloud: An approach based on genetic algorithm and deep reinforcement learning," *Expert Systems with Applications*, vol. 234, p. 120972, 2023.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," *Special Publication 800-145*, National Institute of Standards and Technology, September 2011.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[4] D. Ergenc¸, A. Memedi, M. Fischer, and F. Dressler, "Resilience in edge computing: Challenges and concepts," Foundations and Trends® in Networking, vol. 14, pp. 254–340, May 2025.

[5] Z. Li, N. Zhu, D. Wu, H. Wang, and R. Wang, "Energy-efficient mobile edge computing under delay constraints," IEEE Transactions on Green Communications and Networking, vol. 6, no. 2, pp. 776–786, 2022.

[6] Y. Chiang et al., "Management and orchestration of edge computing for IoT: A comprehensive survey," IEEE Internet of Things Journal, vol. 10, no. 16, pp. 14307–14331, 2023.

[7] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," Journal of Cloud Computing, vol. 6, no. 19, 2017.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proc. of the MCC Workshop on Mobile Cloud Computing, pp. 13–16, ACM, 2012.

[9] S. H. and N. V., "A review on fog computing: Architecture, fog with IoT, algorithms and research challenges," ICT Express, vol. 7, no. 2, pp. 162–176, 2021.

[10] C. Puliafito et al., "Fog computing for the internet of things: A survey," ACM Trans. Internet Technol., vol. 19, no. 2, 2019.

[11] T. Kolajo, O. Daramola, and A. Adebiyi, "Big data stream analysis: A systematic literature review," Journal of Big Data, vol. 6, June 2019.

[12] O.-C. Marcu et al., "Storage and ingestion systems in support of stream processing: A survey," Tech. Rep., Inria, 2018.
[13] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, pp. 30–39, Jan. 2017.
[14] M. Fomichev et al., "No one size (PPM) fits all: Towards privacy in stream processing systems," in Proc. of the 17th ACM Int'l Conf. on Distributed and Event-Based Systems (DEBS '23), pp. 61–67, 2023.
[15] A. T. Atieh, "The next generation cloud technologies: A review on distributed cloud, fog and edge computing and their opportunities and challenges," ResearchBerg Review of Science and Technology, vol. 1, no. 1, pp. 1–15, 2021.
[16] J. Singh, P. Singh, and S. S. Gill, "Fog computing: A taxonomy, systematic review, current trends and research challenges," Journal of Parallel and Distributed Computing, vol. 157, 2021.
[17] M. S. Jawed and M. Sajid, "A comprehensive survey on cloud computing: Architecture, tools, technologies, and open issues," Int'l Journal of Cloud Applications and Computing (IJCAC), vol. 12, no. 1, pp. 1–33, 2022.
[18] P. López et al., "Edge-centric computing," ACM SIGCOMM Computer Communication Review, vol. 45, pp. 37–42, Sept. 2015.
[19] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," IEEE Communications Surveys & Tutorials, vol. PP, pp. 1–1, March 2018.
[20] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: Integration and load balancing in data centers," in SC '08: Proc. of the 2008 ACM/IEEE Conf. on Supercomputing, pp. 1–12, 2008.
[21] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: A load and storage balancing algorithm for distributed multimedia servers," in Proc. IEEE Int. Conf. on Comput. Design (ICCD), pp. 170–174, June 1996.
[22] M. Capra et al., "Edge computing: A survey on the hardware requirements in the internet of things world," Future Internet, vol. 11, no. 4, 2019.
[23] L. Ju, P. Singh, and S. Toor, "Proactive autoscaling for edge computing systems with Kubernetes," in Proc. of the 14th IEEE/ACM Int'l Conf. on Utility and Cloud Computing Companion, 2022.
[24] H. Ahmad et al., "Towards resource-efficient reactive and proactive auto-scaling for microservice architectures," Journal of Systems and Software, vol. 225, p. 112390, 2025.
[25] N.-M. Dang-Quang and M. Yoo, "Deep learning-based autoscaling using bidirectional long short-term memory for Kubernetes," Applied Sciences, vol. 11, no. 9, 2021.
[26] J. Wang and D. Li, "Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing," Sensors, vol. 19, no. 5, p. 1023, 2019.
[27] D. Alsadie, "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: Challenges and prospects," PeerJ Computer Science, vol. 10, June 2024.
[28] E. Karimi, Y. Chen, and B. Akbari, "Task offloading in vehicular edge computing networks via deep reinforcement learning," Computer Communications, vol. 189, pp. 193–204, 2022.
[29] S. Parikh et al., "Security and privacy issues in cloud, fog and edge computing," Procedia Computer Science, vol. 160, pp. 734–739, Jan. 2019.
[30] L. Schmertmann, K. Hartke, and C. Bormann, "CoDTLS: DTLS handshakes over CoAP," IETF Internet-Draft, 2014. Accessed: June 27, 2025.

AUTHOR

Alireza Faghihi Moghaddam: received B.sc from IAU Tehran in Computer Engineering. Currently, he is pursuing his Masters in Computer Science from the Uppsala University, Sweden. His research interests include distributed computing and machine-learning