International Journal of Computer Science and Engineering (IJCSE)
ISSN (P): 2278–9960; ISSN (E): 2278–9979
Vol. 11, Issue 1, Jan–Jun 2022; 285–296

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

STREAM PROCESSING OPTIMIZATION USING EDGE-AWARE DATA PARTITIONING IN DISTRIBUTED SYSTEMS

Sarvesh Kumar Gupta

Consulting Member of Technical Staff, Oracle, Saint Peters, Missouri -63376, USA

ABSTRACT

Today's distributed stream processing platforms are becoming more and more vulnerable in their ability to handle large data streams being generated by IoT devices, edge sensors, industrial surveillance devices, smart cities, and various cloud applications. Conventional data partitioning methods such as hashing-based and round-robin partitioning may have trouble maintaining low latencies and evenly distributing the load in cases where data streams exhibit geographical scattering, workload imbalances, and nonuniform availability of resources. The focus of the research discussed herein is the development of optimal data partitioning approaches which will improve stream processing performance through intelligent data partitioning considering edge proximity and resource heterogeneity. Specifically, this paper will examine edge-aware partitioning as a potential solution to the challenges posed by traditional partitioning approaches. The study uses a simulation-based experimental approach implemented in Apache Flink to compare hash-based, range-based, load-aware, and proposed edge-aware partitioning strategies.The strategies are evaluated using latency, throughput, and network utilization as performance metrics. It becomes apparent from the results that the use of edge-aware data partitioning considerably improves stream processing due to reduced network communication overhead, minimized latencies, effective load balancing, and resource optimization.

KEYWORDS: *Stream Processing, Edge Computing, Edge-Aware Data Partitioning, Distributed Systems, Real-Time Data Analytics, Load Balancing, Data Locality.*

Article History

Received: *28 Apr 2022* | ***Revised:*** *30 Apr 2022* | ***Accepted:*** *30 Jun 2022*