QoS-Aware Self-Adapting Resource Utilisation Framework for Distributed Stream Management Systems

Authors: Tarjana Yagnik, Dr. Feng Chen, Dr. Laleh Kasraian

Presenter: Tarjana Yagnik

De Montfort University, Leicester, UK

Email: tarjana.yagnik@dmu.ac.uk

The Seventh International Conference on Big Data, Small Data, Linked Data and Open Data
ALLDATA 2021
April 18 -22, 2021 - Porto, Portugal
https://www.iaria.org/conferences2021/ProgramALLDATA21.html

Tarjana Yagnik

- Hello and welcome to my presentation of this paper in ALLDATA2021 conference.
- I am Tarjana Yagnik, a PhD student and a lecturer in Computer Science at De Montfort University

- Qualification:
  - BSc
  - MCA (Master in Computer Application)
  - MSc Computing
  - AFHEA (Assistance Fellow of Higher Education Academy)

- Experience:
  - Software developer
  - Web and Server Manager
  - Lecturer in Computer Science

- Interested Area:
  - Traditional to non-traditional DBMS including Distributed database system
  - Software and web application development
  - Programming languages JAVA, C, C++, PHP etc
  - System modelling and System life cycle.
  - Big Data analytics

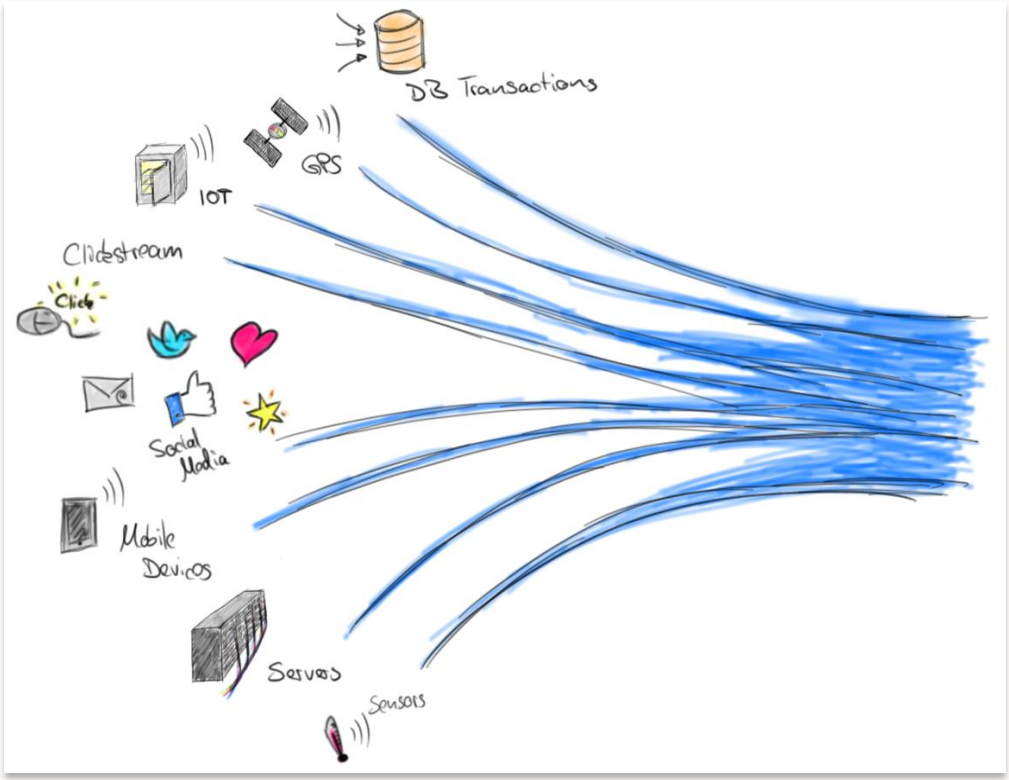
- My research topic is motivated from the limitations of traditional databases that has lead to big data analysis in real-time. I am interested in scheduling strategies of Data Stream Management System, particularly when quality of service requirements are taken into consideration.

Paper Presentation Outline

- Introduction
  - DBMS vs DSMS
  - Quality of Service (QoS) and Resource Allocation
- The Proposed Framework
  - Framework Architectural Design
  - Framework Components
- Implementation and Findings
  - Latency vs Workers
  - Component Profiling and Congestion
  - Metrics Correlation Analysis
- Conclusion and Future Work
- Bibliography

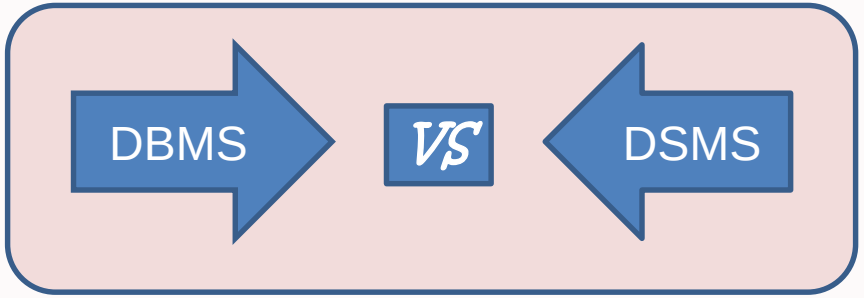
Introduction – DBMS vs DSMS

- During the last decade, a new category of data-intensive applications has emerged and has been recognised by the researchers and industry professionals.
- A data stream is defined as a real-time, continuous, ordered sequence of data items.
- The systems deployed to manage data streams are called Data Stream Management Systems (DSMS).



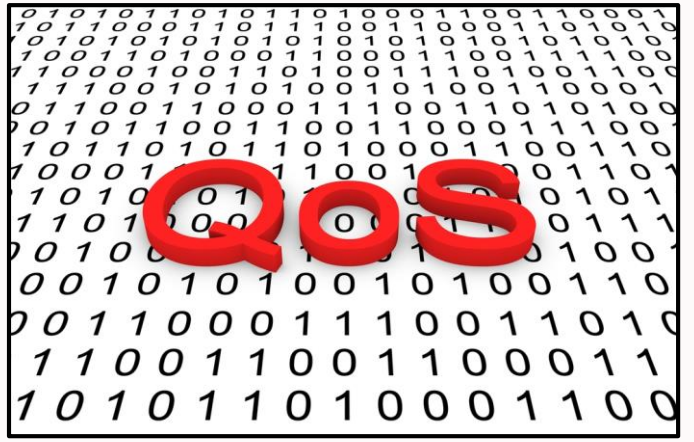
https://trojrobert.github.io/azure-and-aws-data-stream-analytics-and-processing-aws-kinesis,-azure-stream-analytics-and-azure-event-hub/

DSMS is entirely different from the traditional Database Management Systems (DBMS). In the traditional DBMS, the data is stored in the system and queries are run dynamically whereas in the DSMS, queries are submitted persistently and data streams run transiently or dynamically over them.



Quality of Service

- Quality of Service (QoS) such as latency and throughput are important attributes of overall performance of the system.
- Critical applications such as healthcare monitoring, critical infrastructure applications, military command/control applications, it is vital that such strict QoS levels are always met.
- Main challenge:
  - How to deliver the pre-defined QoS requirements efficiently and effectively to the user/application.



Resource Allocation

- Resource Allocation Optimisation is an another issue in DSMS.
- DSMS should have the ability to distribute and allocate physical computing resources between the submitted queries and fulfil the required QoS specification in a fair and square manner.
- The DSMS uses scheduling strategy for resource allocation
- Two main issues while managing resource allocation:
  1. The ability of the system to allocate or release computing resources to meet an application workload and specified quality of service requirements
  2. Devising violations and performing the relevant optimisation actions to alter the system configuration during the runtime.

https://toughnickel.com/business/How-to-Do-Resource-Allocation-in-Excel

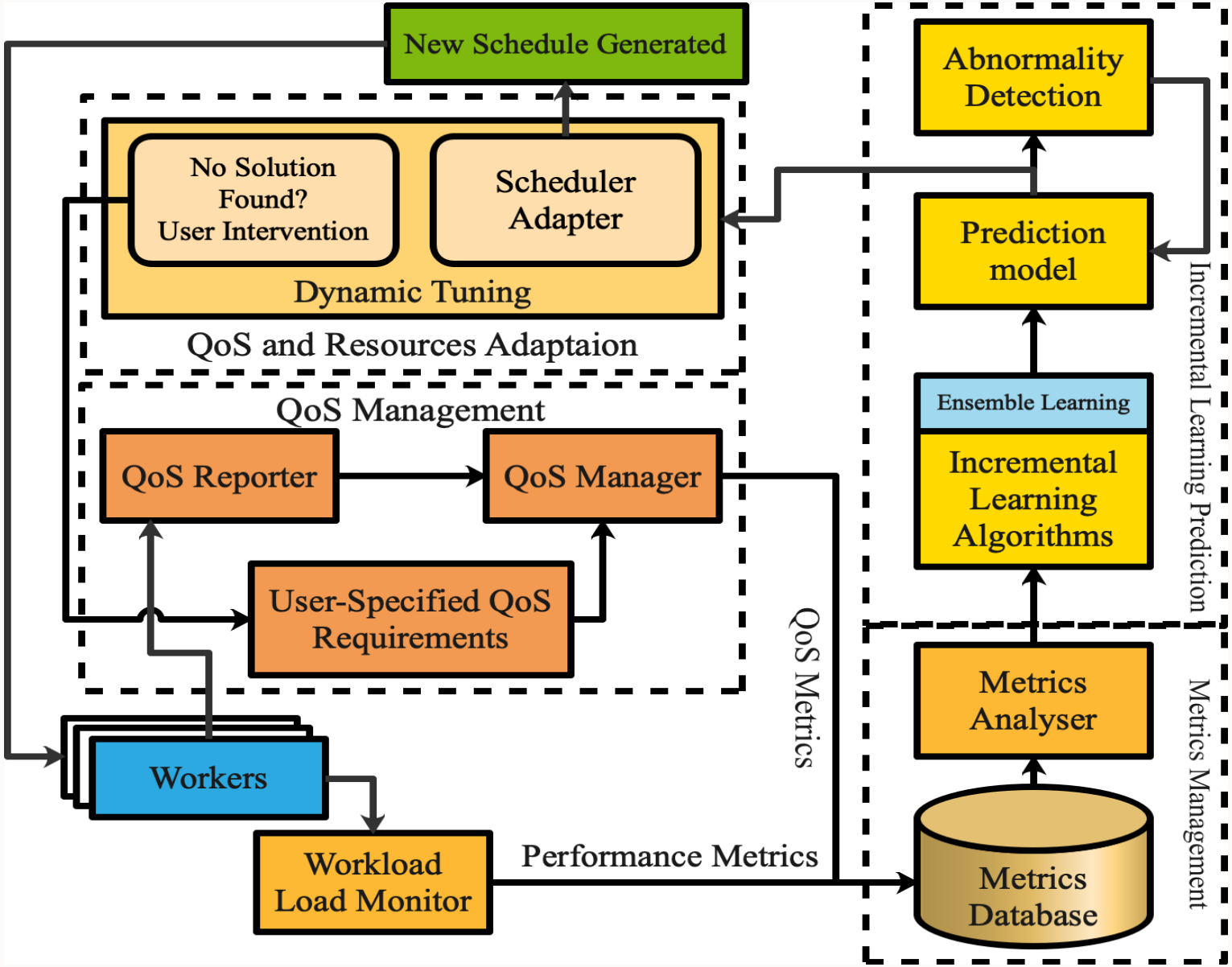
Many frameworks have been proposed for QoS management and resource allocation for the various Distributed Stream Management Systems (DSMS) but lack the capability of dynamic adaptation to fluctuations in input data rates.

The Proposed Framework

- **QoS-Aware Self-Adaptive Resource Utilisation framework** for data stream management systems is presented.
- A comprehensive usage model that comprises mixing of instantaneous reactions with proactive actions is incorporated within the proposed framework.

- **Instantaneous reactions** include applying real-time analytics on collected performance and QoS metrics of each component of the system (worker profiling) before such data becomes obsolete and loses its value.
- **Proactive actions** are the processes of applying predictive models that further assist in decision making and resource allocation planning and scheduling within the system in a real-time streaming environment.

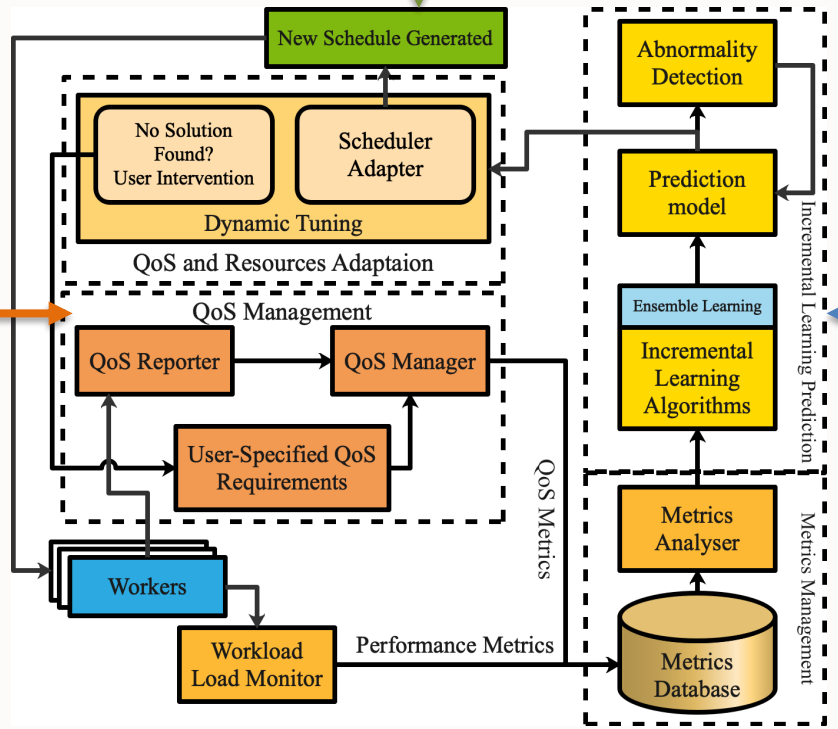
Framework Architectural Design



Four main components of the framework

1. QoS Management

2. Performance Metrics Management



3. Incremental Learning Prediction with Abnormality Detection

4. The Dynamic Tuning / Scheduler Adapter Module

Framework Components Process

1. The QoS Manager:

- It stores the various reports that it receives from its reporters located within the various components within the DSMS. The QoS management is composed of the two elements:
  - User QoS Specifications:
  - System QoS Metrics Collector:

2. **Performance Metrics Management:**

- All performance metrics are measured and associated calculations are done over the specified window of time.
- Aggregated results will be reported through a metrics collection pipeline.
- Certain metrics normalisation factor and correlation analyse are done to:
  - reduce the data dimensions and
  - identify the most relevant set of metrics.

Framework Components Process cont.

3. **Incremental Learning Prediction:**

The DSMS cluster nodes are affected by fluctuations caused by the abnormal data rates, network transmission and other factors such as nodes capacity over the period and QoS violation.

4. **Dynamic Tuning and Scheduler Adapter Module:**

- The Scheduler adapter initiates necessary countermeasures in the form of modifications to the run-time scheduling strategy.
- When the specified QoS constraint are been met, the system configuration is stabilised
- If it doesn't find the solution then the result will be reported to the system/user to decide on the best action to be done
  - Either by adding additional physical resources or
  - relaxing the QoS specifications for this particular query.

Implementation

- To validate the proposed framework and evaluate its performance, the first set of experiments have been conducted using a complete apache storm installation in local mode where all services and daemons are installed in one machine.
  - Storm is a fast, benchmark clocked, open-source, real-time, scalable distributed and fault-tolerant Data Stream Management System.
- The proposed framework is adaptable to any other DSMSs.
- The goal is to collect several performance and QoS metrics through the deployment of the Metrics Collector Module.

Findings: Total Latency vs Number of Workers

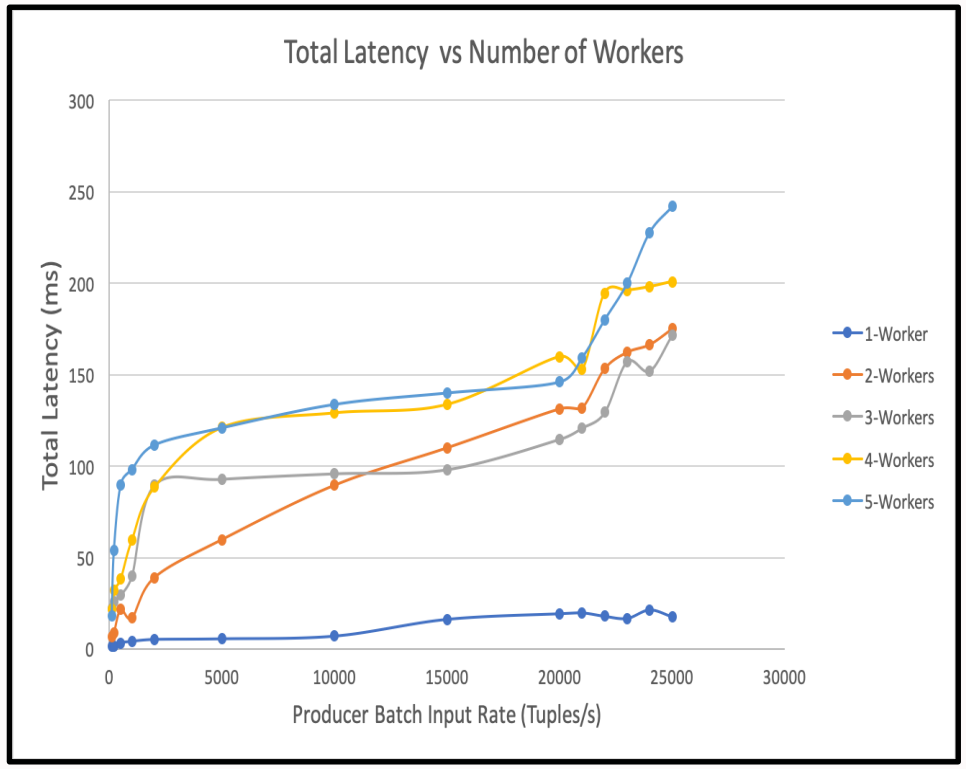


Figure 1: Total Latency of the WordCountTopology running on multiple workers within the same worker node

- The experiments started with input data rates of 100 tuples and increased to 25,000 tuples/second running on one worker and then extended to 5 workers.
- Usually, it states that during the normal operational activities, the available resources are able to handle the fluctuations in data input rate and the latency/throughput within acceptable levels.
- However, from the graph it is visible that QoS metrics are seriously degraded as the input data rates exceeded the computing capacity of the system.
- The reason is, running the topology using several workers introduces traffic between inter-workers and intra-workers which affects the performance as well the QoS attributes considerably.

Findings: Component Profiling and Congestion

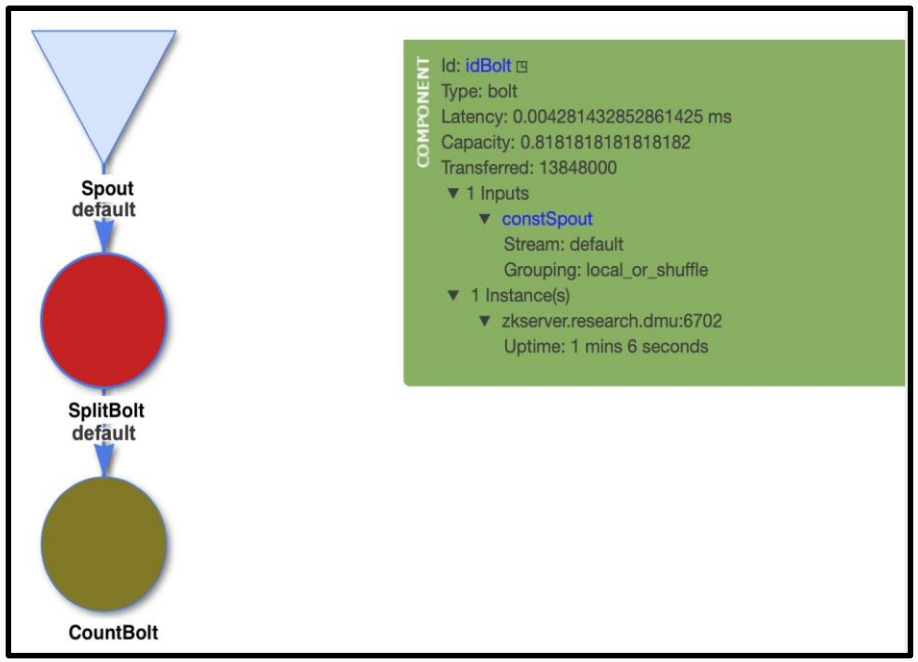


Figure 2. Congestion caused by high input data rates and its effects on the processing capacities of the various topology components

- Monitoring WordCountTopology Storm Web Graphical User Interface.
- Congestion caused by high input data rates that has affected the processing capacities and execution latency of the various topology components.
- Bolt (Split) Capacity 0.81 which is out of the threshold or the acceptable range of QoS.
- This indicates, specific component of the topology is causing congestion where resource reaches its capacity limits so such component need to be examined.

Findings: Correlation Analysis

Spearman's Correlations						
			Spearman's rho	p	Lower 95% CI	Upper 95% CI
rate(tuples/s)	-	median(ms)	0.629***	< 0.001	0.587	0.669
rate(tuples/s)	-	cores	0.675***	< 0.001	0.637	0.71
rate(tuples/s)	-	completed	1.000***	< 0.001	1.000	1.000
rate(tuples/s)	-	user_cpu (ms)	0.798***	< 0.001	0.772	0.821
rate(tuples/s)	-	mean(ms)	0.652***	< 0.001	0.612	0.69
rate(tuples/s)	-	ui_complete_latency (ms)	0.730***	< 0.001	0.695	0.762
Median(ms)	-	cores	0.816***	< 0.001	0.792	0.837
Median(ms)	-	completed	0.630***	< 0.001	0.587	0.669
Median(ms)	-	user_cpu (ms)	0.825***	< 0.001	0.802	0.845

Fig 3: Correlation Table of partial Metrics based on the Spearman's Correlation Coefficients with %95 confidence intervals

Variable	Input rate (tuples/s)	Median (ms)	Cores	Completed	User_cpu (ms)	Latency (ms)	Mean (ms)
Rate (tuples/s)	1.000	0.629	0.675	1.000	0.798	0.652	0.730
Median(ms)	0.629	1.000	0.816	0.630	0.825	0.987	0.746
Cores	0.675	0.816	1.000	0.675	0.960	0.827	0.890
Completed	1.000	0.630	0.675	1.000	0.798	0.653	0.730
User_cpu (ms)	0.798	0.825	0.960	0.798	1.000	0.840	0.944
Mean (ms)	0.652	0.987	0.827	0.653	0.840	1.000	0.768
Latency (ms)	0.730	0.746	0.890	0.730	0.944	0.768	1.000

Fig 4: Heatmap representation of the correlation between the metrics

- Correlation analysis is done to evaluate the strength of relationship between the various metrics collected from the different components of the topologies and under variable operational environments.
- High correlation coefficients mean that two variables have a strong relationship with each other.
- A heat map representation of the correlation relationships is presented and highlights the most relevant metrics that will be used in the future experiments.
- The vulnerable components can be identified and can be replaced with better computing resources or increase the number of executors allocated for that particular component.

Conclusions and Future Work

- Conclusions
  - Established the importance of QoS and Resource Allocation.
  - Proposed QoS-Aware Self-Adapting Resource Utilisation framework.
  - Explained the process of each components of the framework.
  - Presented implementation details and findings
- Future Work
  - The rest of the experiments of this research will be carried out using computing instances from Google Cloud Computing Platform to simulate the real environment and fully validate the applicability and performance gains of the proposed framework.

Bibliography

1. A. Arasu, et al., “Stream: The stanford data stream management system,” In Data Stream Management, Springer, Berlin, Heidelberg, 2016, pp. 317-336.
2. A. Muhammad, and M. Aleem, “A3-Storm: topology-, traffic-, and resource-aware storm scheduler for heterogeneous clusters,” JOURNAL OF SUPERCOMPUTING, vol. 77, pp. 1059-1093, May 2020.
3. “Apache Storm,” Apache Software Foundation, 2014. [Online]. Avail- able: https://storm.apache.org/about/integrates.html. [Accessed 08 April 2021].
4. D. J. Abadi, et al., “Aurora: a new model and architecture for data stream management,” the VLDB Journal, vol. 12(2), pp.120-139, 2003.
5. M. Garofalakis, J. Gehrke, and R. Rastogi, “Data stream management: A brave new world,” In Data Stream Management, Springer, Berlin, Heidelberg, 2016, pp. 1-9.
6. M. HoseinyFarahabady, A. Y. Zomaya, and Z. Tari, “QoS-and contention-aware resource provisioning in a stream processing engine,” In 2017 IEEE International Conference on Cluster Computing (CLUSTER), September 2017, pp. 137-146.
7. N. Tantalaki, S. Souravlas, M. Roumeliotis, and S. Katsavounis, “Pipeline-Based linear scheduling of big data streams in the cloud,” IEEE Access, vol. 8, pp.117182-117202, June 2020.
8. Y. Wei, V. Prasad, S. H. Son, and J. A. Stankovic, “Prediction-based QoS management for real-time data streams,” In 2006 27th IEEE International Real-Time Systems Symposium (RTSS’06), IEEE, December 2006, pp. 344-358.

tarjana.yagnik@dmu.ac.uk; tarjanay@gmail.com