Pratyush Agnihotri. Autonomous Resource Management in Distributed Stream Processing Systems. In the Proceedings of the 22nd International Middleware Conference: Doctoral Symposium, pp. 19-22, ACM press

Autonomous Resource Management in Distributed Stream Processing Systems

Pratyush Agnihotri
Technical University of Darmstadt
Darmstadt, Germany
pratyush.agnihotri@kom.tu-darmstadt.de

ABSTRACT

Resource management in Distributed Stream Processing Systems (DSPS) defines the way queries are deployed on resources to deliver query result while fulfilling the Quality of Service (QoS) requirements of the end-users. Various resource management mechanisms have been proposed in DSPS; however, they become inefficient in challenging conditions imposed by the dynamic environment and heterogeneous resources. Existing works focus on pre-configuration of distinct QoS requirements which cannot be satisfied in case of workload drift. In addition, they lack cooperation between heterogeneous resources leading to inconsistent and incorrect query result. To solve the above challenges, we propose mechanisms i) to forecast the performance of network and heterogeneous resources, ii) to select efficient resource management approach, and iii) for cooperation between resources in dynamic environment.

CCS CONCEPTS

• **Computer systems organization → Real-time systems**; • **Information systems → Stream management**.

KEYWORDS

Stream processing, Learned resource management

1 INTRODUCTION

Social media platforms are an essential part of the Internet of Things (IoT) ecosystem which has a huge users base to disseminate information in real-time. With the surge and gradual increase of IoT devices, social media has become large-scale, easy to use open stage platform to share and disseminate knowledge, thoughts, and ideology. In recent times, malefactors has frequently exploited social media to propagate fake news and cause social and economic instability leading to Infodemic and information pollution [30, 45]. Fake news on social media has questioned the credibility and trustworthiness of the content available online. Therefore, timely detection of the fake content on cross social media platforms presents significant challenges and requires attention of researchers. For example, the fake news on Twitter about explosion at the White House spreads in seconds which resulted in 10 billion USD loss to stock market [30].

In this context, DSPS is a well-known paradigm to process streaming data on arrival to derive high-level events (e.g., fake news detection in social media) while ensuring QoS requirements of the applications (e.g., time criticality to detect fake news) [9]. Resource management mechanisms in DSPS can be utilized for fake news

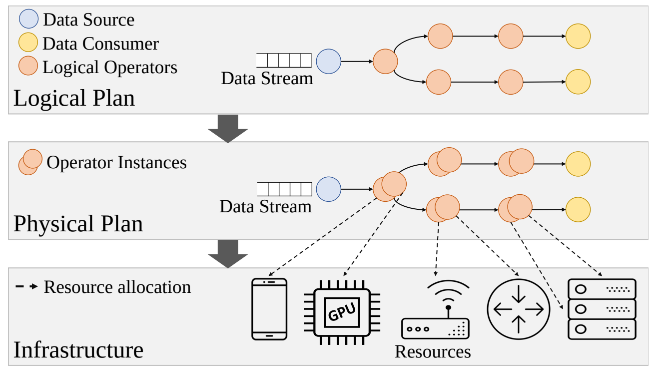

Figure 1: Resource Management in DSPS

detection by deploying and resolving the query on distributed resources. In this work, we focus on two core mechanisms of resource management: *parallelization* (how to parallelize operators) and *elasticity* (how to add and remove resources) mechanisms to deal with the dynamics of environment (e.g., changing workload) fig. 1. In recent years, different resource management mechanisms have been proposed in DSPS [6, 8, 9, 22, 35]. Most of the existing mechanisms on parallelization and elasticity either rely on optimization [3, 15, 16, 19] or machine learning based methods [26, 41, 46]. However, these works cannot meet the latency requirements of time-critical applications such as fake news detection. In addition, they cannot deal with unseen behavior of the dynamic environment and rely on pre-configuration of one or two performance metrics. Some mechanisms utilize edge devices [20, 41] to deliver low latency; however, they suffer due to limited processing capability and lack of complete data. Few approaches perform distributed processing using a combination of edge and cloud/serverless computing [4, 24, 25]. However, such a parallel processing of data often leads into inconsistent and inaccurate result.

In contrast, we propose to consider unseen conditions imposed by dynamic and heterogeneous environment as a primary challenge for designing our autonomous resource management mechanism for DSPS. Our mechanism will select efficient resource management approach at run-time based on the forecast of the performance of network and heterogeneous resources. In addition, we propose a mechanism to ensure cooperation between heterogeneous resources to ensure in-order and consistent delivery of events.

The rest of the paper is structured as follows. Section 2 presents our motivation and research challenges in realizing resource management at distributed infrastructure. Then Section 3 provides an overview of research questions and possible approach to solve these

Pratyush Agnihotri. Autonomous Resource Management in Distributed Stream Processing Systems. In the Proceedings of the 22nd International Middleware Conference: Doctoral Symposium, pp. 19-22, ACM press

challenges. Next, Section 4 presents existing resource management mechanisms. Finally, we will conclude in Section 5.

2 MOTIVATING USE CASES

Social media has evolved into a large-scale and easy to use open platform for sharing and disseminating knowledge and ideology. Malefactors exploit social media to propagate fake news, causing social and economic instability often termed as Information pollution and Infodemic [45]. Therefore, the timely detection of fake news presents a significant challenge to the credibility of online content and requires a major attention to drive research [30, 40].
C1: Existing machine learning [1, 12] and deep learning techniques [11, 37] for fake news detection suffer from workload drift and hence cannot generalize for unseen workload.
C2: Social media platforms generate high volume of data that varies across the heterogeneous devices depending upon social, geographical and temporal context, e.g., high volume of tweets at the time of election [7].
C3: Social media content is generated by resource-constrained devices such as smartphones and commodity hardware like laptop. Local processing of this enormous data for fake news detection is inefficient due to the lack of processing capability and unavailability of complete data. Hence, the detection has to be done in efficient way, i.e., in low latency manner and at the same time dealing varying workload from heterogeneous platforms.
C4: Since local processing for fake news detection can lead to inefficiency, distributed processing can be utilized for this purpose. However, this could lead to inaccuracy in the detection. Therefore, it is important to maintain consistency and order to ensure there is no inaccuracy of data.

3 RESEARCH QUESTIONS

Considering the limitations of existing resource management mechanisms such as operator parallelization and elasticity under heterogeneous and dynamic environment, e.g., social media analytics (cf. Section 2). We have identified the following research questions and gaps in existing work that we will solve in this research.

RQ1: How to forecast the performance of the network and heterogeneous resources under dynamics of environmental conditions? (C1 and C2)[1]
Existing approaches for resource management focus on i) monitoring and sampling techniques [14, 21] to determine the performance and ii) they target a limited number of performance metrics such as average processing latency per operator instance [23], idle and execution time of operator [44], and the resource usages of the operator and its instances [21]. A few approaches target performance evaluation using machine learning techniques such as using Gaussian process [46], incremental learning [43], and using mathematical models such as Game theory [16], Queueing theory [20] in operator parallelization and elasticity. However, these approaches are workload-specific and hence cannot handle the dynamics of the workload and environment. In contrast, we propose a performance model that considers multiple performance metrics while dealing with the dynamics of the environment, such as changing workload and queries. In this research, we will evaluate learning-based methods such as Distributed Deep Learning [13], Reinforcement Learning [27] and propose an approach inspired by these methods to deal with the challenges as mentioned above.

RQ2: When and how to dynamically adjust the parallelism degree at run time in response to the continuous variations of workload based on the forecast of performance? (C2)
Existing literature presents mechanisms for adjusting the parallelism degree for resource management using monitoring and controlling, such as using rule-based approach [18], where parallelism adjustment relies on the change in pre-defined threshold values of performance metric. Other works are centralized and decentralized parallelism adjustment techniques based on Queueing theory [3, 10, 28], and Game theory [31]. These works do not consider unseen behavior of the dynamic environment and rely on pre-configuration of one or two performance metrics to adjust parallelism in a homogeneous resources environment. In contrast, we consider multiple performance metrics for our parallelism adjustment model. Hence, in this model we dynamically adapt the parallelism degree of operators based on the changes in the environment for time-critical applications.

RQ3: How to automatically identify a parallelism mechanism at run time to ensure the QoS requirements of the end-users? (C2 and C3)
Existing literature consists of parallelism mechanisms in two directions i) task- and pipeline-based [17, 32, 42], and ii) data-based parallelism [5, 29, 38]. Each parallelism mechanism is designed to optimize a specific set of performance metrics. The end goal is to meet the QoS requirements of the end-users. For example, Khandekar et al. [17] propose a mechanism based on task and pipeline parallelism to balance CPU capacity and communication cost. While, Schneider et al. [38] used shuffle grouping-based data parallelization for load balancing between operator instances with varying throughput. Backman et al. [2] present a scheduling framework using both data- and task-based parallelism to optimize end-to-end latency. There is no single parallelism mechanism that fits for all possible performance metrics. Therefore, we propose a selection algorithm that will select parallelism mechanism based on the given QoS requirements of the end-users. The selection algorithm will take as an input the measured performance metrics by the forecast model (RQ1) for selection as well as the parallelism degree (RQ2) for auto-scaling.

RQ4: How parallel operator and its instances will cooperate with other operators in an operator graph to ensure order and consistency of the complex events? (C3 and C4)
In parallelism mechanisms, either multiple operator instances execute in parallel (data-based) or input streams are replicated (task-based). Existing mechanisms use a *split and merge* approach for data-based parallelism and a *multiply and merge* approach for task-based parallelism. Parallelism mechanism can choose different types of splitter and multiplier to process input stream and merger to produce output data stream [9, 35]. For example, a key-based splitter in data-based parallelism relies on an input stream with respective key-values [34]. In the case of dynamic change in network traffic and input stream, it might be the case that keys are not evenly distributed in an input stream, resulting in unequal load on operator instances. In addition, if operator instances have limited capacity to process input stream then the reliability of the output data

[1]This research question targets challenges C1 and C2 specified in Section 2.

stream will be reduced as well. Because of this, resulting output stream could consist of duplicated, inconsistent, and inaccurate result, which will degrade the overall performance of the DSPS system. Hence, while applying parallelism mechanisms, special care is needed to produce the resulting output stream in an ordered and consistent way. Therefore, we propose to investigate different forms of cooperation and strategies that operators can perform to ensure the quality of complex events while ensuring QoS in an unseen behavior of a dynamic environment.

4 RELATED WORK

We have divided existing resource management mechanisms for DSPS into - *optimization*- and *machine learning*- approaches.

Optimization techniques. Researchers have intensively applied various optimization techniques based on heuristics [15, 39], integer linear programming [3], rules [19], gaming theory [16, 33], queueing theory [20, 28] for optimal decision-making to manage resources. However, they focus on either fixed QoS requirement or only one or two performance metrics, e.g., heuristic-based parallelization provides better response time and bandwidth in a decentralized network [39], two-step hyper-heuristic model is applied to reduce execution time and energy consumption [15], etc. In addition, optimization models are often combined with other models to improve efficiency and accuracy. For example, combining gaming and queueing theory for resource management on heterogeneous edge resources using social awareness of resource as context [20].

Machine learning. Recent advancements in machine learning motivated the researchers to apply them in DSPS for resource management. For example, applying genetic learning for resource management and its adaptation to deal with changing QoS requirements at run-time [25] and service-based resource management in fog environment [41]. Similarly, Gaussian process is used to analyze historical data of workload and latency to adjust the degree of parallelism [46] and reinforcement learning is used to design multi-tier resource provisioning mechanism [36]. However, these approaches rely on pre-configuration of single or multiple performance metrics and cannot deal with unseen workload.

5 CONCLUSION

Social media platforms are losing content credibility and trustworthiness due to their frequent usage to propagate fake news. Thus, it becomes critically important to detect fake news timely to compensate for the severe damage. In this context, resource management mechanisms in DSPS can process a high volume of data; however, these mechanisms are inefficient in a dynamic and heterogeneous environment. Our research work aims to investigate and propose mechanisms to forecast the performance of the network and heterogeneous resources under dynamics of environmental conditions. This performance forecast will enable selecting a suitable resource management mechanism at run-time to manage in-network heterogeneous resources such as cloud, fog, and edge efficiently. In addition, we propose a mechanism to enable cooperation between in-network resources to ensure in-order and consistent delivery of events.

ACKNOWLEDGEMENT

The author would like to thank his supervisors Prof. Dr. Boris Koldehofe, Dr.-Ing. Thomas Tregel, Prof. Dr.-Ing. Ralf Steinmetz. This work has been co-funded by the German Research Foundation (DFG) as part of the project C2 within the Collaborative Research Center (CRC) 1053– MAKI.

REFERENCES

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy* 1 (2018), 1–9.
[2] Nathan Backman, Rodrigo Fonseca, and Uğur Çetintemel. 2012. Managing parallelism for stream processing in the cloud. In *Proceedings of the 1st International Workshop on Hot Topics in Cloud Data Processing (HotCDP).* 1–5.
[3] Valeria Cardellini, Vincenzo Grassi, Francesco Lo Presti, and Matteo Nardelli. 2017. Optimal operator replication and placement for distributed stream processing systems. *ACM SIGMETRICS Performance Evaluation Review* 44 (2017), 11–22.
[4] Valeria Cardellini, Francesco Lo Presti, Matteo Nardelli, and Gabriele Russo Russo. 2018. Decentralized self-adaptation for elastic data stream processing. *Future Generation Computer Systems* 87 (2018), 171–185.
[5] Gianpaolo Cugola and Alessandro Margara. 2012. Low latency complex event processing on parallel hardware. *J. Parallel and Distrib. Comput.* 72 (2012), 205–218.
[6] Gianpaolo Cugola and Alessandro Margara. 2012. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)* 44 (2012), 1–62.
[7] Kareem Darwish, Walid Magdy, and Tahar Zanouda. 2017. Trump vs. Hillary: What went viral during the 2016 US presidential election. In *International Conference on Social Informatics (ICSI).* 143–161.
[8] Miyuru Dayarathna and Srinath Perera. 2018. Recent advancements in event processing. *ACM Computing Surveys (CSUR)* 51 (2018), 1–36.
[9] Marcos Dias de Assuncao, Alexandre da Silva Veith, and Rajkumar Buyya. 2018. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. *Journal of Network and Computer Applications* 103 (2018), 1–17.
[10] Tiziano De Matteis and Gabriele Mencagli. 2017. Elastic scaling for distributed latency-sensitive data stream operators. In *Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP).* 61–68.
[11] Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended overview of CLEF HIPE 2020: named entity processing on historical newspapers. In *CEUR Workshop Proceedings.*
[12] Elshrif Elmurngi and Abdelouahed Gherbi. 2017. Detecting fake reviews through sentiment analysis using machine learning techniques. *IARIA/data analytics* (2017), 65–72.
[13] Benjamin Hilprecht, Carsten Binnig, and Uwe Röhm. 2019. Towards learning a partitioning advisor with deep reinforcement learning. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management (aiDM).* 1–4.
[14] Gabriela Jacques-Silva, Buğra Gedik, Rohit Wagle, Kun-Lung Wu, and Vibhore Kumar. 2012. Building user-defined runtime adaptation routines for stream processing applications. *arXiv preprint arXiv:1208.4176* (2012).
[15] Sabihe Kabirzadeh, Dadmehr Rahbari, and Mohsen Nickray. 2017. A hyper heuristic algorithm for scheduling of fog networks. In *Proceedings of the 21st Conference of Open Innovations Association (FRUCT).* 148–155.
[16] Junaid Ahmed Khan, Cedric Westphal, and Yacine Ghamri-Doudane. 2017. Offloading content with self-organizing mobile fogs. In *29th International Teletraffic Congress (ITC).* 223–231.
[17] Rohit Khandekar, Kirsten Hildrum, Sujay Parekh, Deepak Rajan, Joel Wolf, Kun-Lung Wu, Henrique Andrade, and Buğra Gedik. 2009. COLA: Optimizing stream processing applications via graph partitioning. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing.* 308–327.
[18] Roland Kotto Kombi, Nicolas Lumineau, and Philippe Lamarre. 2017. A preventive auto-parallelization approach for elastic stream processing. In *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS).* 1532–1542.
[19] Changlong Li, Hang Zhuang, Qingfeng Wang, and Xuehai Zhou. 2018. SSLB: Self-Similarity-Based Load Balancing for Large-Scale Fog Computing. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)* 43, 12 (2018).
[20] Liqing Liu, Zheng Chang, and Xijuan Guo. 2018. Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices. *IEEE Internet of Things Journal* 5 (2018), 1869–1879.

[21] Xunyun Liu and Rajkumar Buyya. 2017. Performance-oriented deployment of streaming applications on cloud. *IEEE Transactions on Big Data* 5 (2017), 46–59.

[22] Xunyun Liu and Rajkumar Buyya. 2020. Resource management and scheduling in distributed stream processing systems: a taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)* 53 (2020), 1–41.

[23] Xunyun Liu, Amir Vahid Dastjerdi, Rodrigo N Calheiros, Chenhao Qu, and Rajkumar Buyya. 2017. A stepwise auto-profiling method for performance optimization of streaming applications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 12 (2017), 1–33.

[24] Manisha Luthra, Sebastian Hennig, Kamran Razavi, Lin Wang, and Boris Koldehofe. 2020. Operator as a Service: Stateful Serverless Complex Event Processing. In *Proceedings of the IEEE International Conference on Big Data (BigData)*. 1964–1973.

[25] Manisha Luthra, Boris Koldehofe, Niels Danger, Pascal Weisenberger, Guido Salvaneschi, and Ioannis Stavrakakis. 2021. TCEP: Transitions in operator placement to adapt to dynamic network environments. *J. Comput. System Sci.* 122 (2021), 94–125.

[26] Manisha Luthra, Boris Koldehofe, Pascal Weisenburger, Guido Salvaneschi, and Raheel Arif. 2018. TCEP: Adapting to dynamic user environments by enabling transitions between operator placement mechanisms. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems (DEBS)*. 136–147.

[27] Ruben Mayer and Hans-Arno Jacobsen. 2020. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. *ACM Computing Surveys (CSUR)* 53 (2020), 1–37.

[28] Ruben Mayer, Boris Koldehofe, and Kurt Rothermel. 2014. Meeting predictable buffer limits in the parallel execution of event processing operators. In *Proceedings of the IEEE International Conference on Big Data (BigData)*. 402–411.

[29] Ruben Mayer, Boris Koldehofe, and Kurt Rothermel. 2015. Predictable low-latency event detection with parallel complex event processing. *IEEE Internet of Things Journal* 2 (2015), 274–286.

[30] Priyanka Meel and Dinesh Kumar Vishwakarma. 2020. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications* 153 (2020), 112986.

[31] Gabriele Mencagli. 2016. A game-theoretic approach for elastic distributed data stream processing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 11 (2016), 1–34.

[32] Yugo Nakamura, Hirohiko Suwa, Yutaka Arakawa, Hirozumi Yamaguchi, and Keiichi Yasumoto. 2016. Design and implementation of middleware for iot devices toward real-time flow processing. In *Proceedings of the 36th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*. 162–167.

[33] Duong Tung Nguyen, Long Bao Le, and Vijay Bhargava. 2018. Price-based resource allocation for edge computing: A market equilibrium approach. *IEEE Transactions on Cloud Computing* 9 (2018), 302–317.

[34] Nicoló Rivetti, Leonardo Querzoni, Emmanuelle Anceaume, Yann Busnel, and Bruno Sericola. 2015. Efficient key grouping for near-optimal load balancing in stream processing systems. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems (DEBS)*. 80–91.

[35] Henriette Röger and Ruben Mayer. 2019. A comprehensive survey on parallelization and elasticity in stream processing. *ACM Computing Surveys (CSUR)* 52 (2019), 1–37.

[36] Gabriele Russo Russo, Matteo Nardelli, Valeria Cardellini, and Francesco Lo Presti. 2018. Multi-level elasticity for wide-area data streaming systems: A reinforcement learning approach. *Algorithms* 11 (2018), 134.

[37] Saad Sadiq, Nicolas Wagner, Mei-Ling Shyu, and Daniel Feaster. 2019. High dimensional latent space variational autoencoders for fake news detection. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. 437–442.

[38] Scott Schneider, Martin Hirzel, Bugra Gedik, and Kun-Lung Wu. 2012. Auto-parallelizing stateful distributed streaming applications. In *Proceedings of the 21st international conference on Parallel architectures and compilation techniques (PACT)*. 53–64.

[39] Mennan Selimi, Llorenç Cerdà Alabern, Felix Freitag, Luís Veiga, Arjuna Sathiaseelan, and Jon Crowcroft. 2019. A lightweight service placement approach for community network micro-clouds. *Journal of Grid Computing* 17 (2019), 169–189.

[40] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10 (2019), 1–42.

[41] Olena Skarlat, Matteo Nardelli, Stefan Schulte, Michael Borkowski, and Philipp Leitner. 2017. Optimized IoT service placement in the fog. *Service Oriented Computing and Applications* 11 (2017), 427–443.

[42] William Thies, Michal Karczmarek, and Saman Amarasinghe. 2002. StreamIt: A language for streaming applications. In *Proceedings of the International Conference on Compiler Construction (CC)*. 179–196.

[43] Chunkai Wang, Xiaofeng Meng, Qi Guo, Zujian Weng, and Chen Yang. 2017. Automating characterization deployment in distributed data stream management systems. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017), 2669–2681.

[44] Daniel Warneke and Odej Kao. 2011. Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. *IEEE transactions on parallel and distributed systems* 22 (2011), 985–997.

[45] World Health Organization (wHO). 2021. Infodemic. https://www.who.int/health-topics/infodemic. [Online; accessed 12-09-2021].

[46] Nikos Zacheilas, Vana Kalogeraki, Nikolas Zygouras, Nikolaos Panagiotou, and Dimitrios Gunopulos. 2015. Elastic complex event processing exploiting prediction. In *IEEE International Conference on Big Data (BigData)*. 213–222.