Review on Data Stream Partitioning and Re-optimization using Runtime Dependency Mining

Mr. Shinde Sumit S.
Department of Computer Engineering,
DPCOE, Wagholi,
Pune, India

Prof. Joshi Shubham
Department of Computer Engineering,
DPCOE, Wagholi,
Pune, India

***Abstract*— Data stream partitioning is the profitable approach used for distributed data. Distributed data stream partitioning plays important role for a wide range of applications. Applications involved in distributed and web-based systems can inculcate data stream partitioning and processing it gain higher throughput. This can be done using values of specific attributes or by using partitioning keys. Re-partitioning the intermediary streams of data stream is required when different partitioning keys are applied to different queries. This process cause extra communication overhead and in turn reduce throughput. By finding dependencies between partitioning keys applicable for each query, re-partitioning can be reduced. Current system examines query syntax at compile-time, detects inter-key dependencies and in turn avoids re-partitioning. Compile-time methods can be extended by a scalable and elastic stream engine, which process a large data stream. Elasticity feature will help to balance dynamic load. Mining of temporal approximation dependency is applicable for moving time window and it is approximately valid over it. Run-time re-optimization solution dynamically adjusts data processing and re-partitioning used to achieve high throughput without using unnecessary partitioning.**

Keywords*— Data stream processing, Auto-parallelization, Runtime optimization, elasticity, Temporal approximation dependency.

INTRODUCTION

Existence of large class of newly emerging applications, data generated due to external environment is forwarded asynchronously to servers. Server provides services to the data streams. Some sample examples are share-market, banking, location-tracking, sensor networks. These applications need timely processing data stream with great response. Streaming applications are directed graphs where vertices points to operators of query and edges of graph points data streams. Executing expensive continuous queries over bulk data streams requires partitioning keys enables such customized streams.

Auto-parallelization technique involves locating of a region in the applications data flow graph which can be replicated for applying data partitioning at run-time stated by Gedik et.al.[2]. An elastic auto-parallelization dynamically avails nodes at run-time. Without wasting time in re-partitioning of data streams, it provides higher throughput to achieve high elasticity and quick response .Gulisan et.al.[3] presents a computing paradigm based on stream processing engines(SPEs). SPEs are specially designed computing systems for presenting continuous data streams with minimum delay. This work extends present compile-time optimization methods using the methods of runtime re-optimization and dependency mining.

To check out dependencies at run-time temporal approximation dependencies is used. Temporal approximation dependency (TADs), is an approximate dependencies between moving entities over a time window defined by Viel et.al.[1]. TAD discovers new strategies useful for less-partitioning, benefited for reduced communication and higher throughput.

I. PROBLEM DEFINITION

To Design and implement an auto-parallelization data-stream engine facilitated with run-time repartitioning optimization.Which will scale up the data stream processing and enhance the throughput of the data stream processing system.The partitioning strategy of a program strongly affects its performance. A strategy that allocates different partitioning keys to two successive queries will require re-partitioning the intermediate stream between the queries, causing extra communication.

II. PERSPECTIVE SOLUTION

As the compile-time query analysis and optimization is static in nature and it fail to achieve dynamic dependencies valid in a specific time window, Run-time dynamic query analysis and optimization methods play useful role. To achieve this result temporal approximation dependency checking strategy is used. It helps to improve system throughput and system scalability.

III. RELATED WORK AND LITERATURE SURVEY

The need to gather, process and analyze data stream is being increased to get some valuable insights. This is needed to be performed at real-time during processing. Data stream processing with parallelization techniques come from earlier research.

The query aware data stream partitioning approach defined by Theodore Johnson et.al.[8] has specified methods for analyzing any given query node to determine a partition strategy and choosing optimal partitioning, which minimizes overall communication costs. Mitch Chreniack et.al.[4] has described two stream processing systems namely Aurora and Medusa, designed to solve architectural challenges in large-scale distributed stream processing systems. Daniel J. Abadi et.al.[6] specified a new stream processing engine Borealis, which extends both Aurora and Medusa system with advanced capabilities required by newly emerging stream processing applications.

Vincenzo Gulisano et.al.[3] presented a scalable and elastic stream processing engine to process large volume data streams. It uses parallelization techniques to divide single query into sub queries and allocate them to independent nodes. It provides scalability and elasticity to the distributed system.

BurgaGedik et.al.[2] proposed an elastic auto-parallelization solution that can dynamically adjust number of processing channels. Auto-parallelization technique is used to scale up stream processing applications. EmericViel et.al.[1] extended existing compile-time methods based on temporal approximation dependencies between partitioning keys required to partition data stream.

IV.PROPOSED WORK

Implementation of scalable and elastic processing system provides run-time re-optimization using temporal approximation dependency mining with auto-parallelization strategy.For distributed data stream processing, a program made of multiple queries can be parallelized by partitioning input streams according to the values of specific attributes, or partitioning keys.

V.PROPOSED ARCHITECTURE/PROTOTYPE

A. Date Stream Partitioning

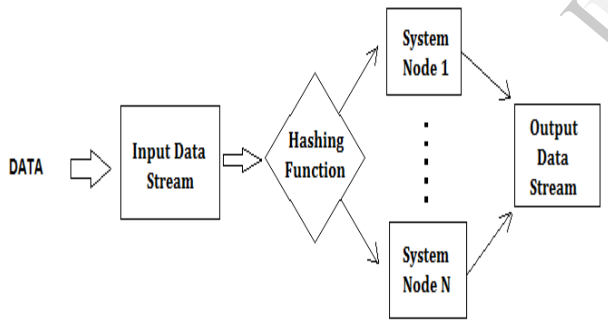

Fig.1. Basic Block Diagram for Data Stream Partitioning

Input data stream is partitioned using hashing function or by using partitioned keys. Partitioned stream is then forwarded to respective system node for processing.

B. Distributed Architecture:

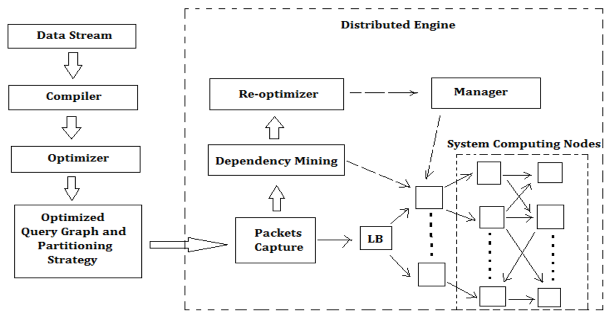

Fig.2. Distributed System Architecture

VI.SCOPE OF WORK

Presented Partitioning optimization method extends existing compile-time optimization methods by using run-time re-optimization, based on runtime dependency mining. It adds scalability and elasticity in the distributed processing engine. Introducing the concept of auto-parallelization defined by Gedik et.al.[2] facilitate distributed computing nodes to locate the regions in the applications data flow graph, that can be replicated at run-time to apply data partitioning, in order to achieve scalability.

VII.DISCUSSION

As the work is at very early stage, complete integration with distributed engine required for query and data stream execution, involving implementation of the exception handling and runtime strategy switching modules, as stated by Viel et.al.[1]. Evaluation of throughput and scalability of stream processing engine also needs a real-life evaluation of temporal approximation dependency mining algorithm.

VIII.REFERENCES

1. EmericViel, Haruyasu Ueda, "Data Stream Partitioning Re-Optimization Based on Runtime Dependency Mining", 978-1-4799-3481-2/14/ © 2014 IEEE
2. Bug˘ raGedik, Scott Schneider, Martin Hirzel, and Kun-Lung Wu," Elastic Scaling for Data Stream Processing",Ieee Transactions On Parallel And Distributed Systems, Vol. 25, No. 6, June 2014,pp.1447-1463
3. V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, P. Valduriez, "StreamCloud: An Elastic and Scalable Data Streaming System", IEEE Trans. Parallel Distrib. Syst., vol. 23, pp. 2351-2365,.2012
4. M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, and S. B.Zdonik, "Scalable distributed stream processing", in Proc. CIDR'03, 2003, pp. 257-268.
5. S. Schneider, M. Hirzel, B. Gedik, K. L. Wu, "Auto-parallelizing statefuldistributed streamingApplications", in Proc. PACT'12, 2012, pp. 53-64.
6. D.Abadi, Y.Ahmad,M. Balazinska, U.C¸ etintemel, M.Cherniack J.- Hwang, W.Lindner,A.Maskey, A.Rasin, E. Ryvkina,N. Tatbul, Y. Xing, and S. Zdonik, ‘‘The Design of the Borealis Stream Processing Engine,’’ in Proc. CIDR, 2005, pp. 277-289.
7. Patrik G. Clark, Jerzy W. Grzymala-Busse,"A comparison of Global and Local Probabilistic Approximations in Mining Data with Many Missing Attribute Value", IEEE International Conference on Granular Computing(GrC), 978-1-4799-1282-7/13

8. T. Johnson, M. S. Muthukrishnan, V. Shkapenyuk, and O. Spatscheck, "Query-aware partitioning for monitoring massive network data streams", in Proc. SIGMOD'08, 2008, pp. 1135-1146.
9. H. Kurihara, H. Ueda, S. Sakamoto, and M. Matsubara, "Development and runtime platform and high-speed processing technology for data utilization", FUJITSU Sci. Tech. J., Vol 50, No. 1, Jan 2014.
10. E. Zeitler and T. Risch. "Massive scale-out of expensive continuous queries", in Proc.VLDBEndowment, vol. 4, pp. 1181-1188, Aug. 2011.
11. A. Arasu, M. Cherniack, E. F. Galvez, D. Maier, A. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibetts, "Linear road: a stream data management benchmark", in Proc. VLDB'04, 2004, pp. 480-491.

VIII.Author Bibliography

	Mr. Shinde Sumit Sudhakar He has completed Bachelor of Engineering in Information Technology from Amravati University. Currently pursuing Master of Engineering from Pune University.
	Prof. Shubham Joshi He has completed Bachelor of Engineering in Computer Engineering and Master of Engineering in Information Security. Currently He is pursuing Ph.D. He has published 22 research papers, 01 Book. He is Microsoft certified professionaland Chairpersonof Publicity & Web Hosting, IEEE MP Subsection, Reviewer of the International Conferences and Journals.Woking as a PG Co-ordinator at DPCOE,Pune.