


# Optimizing Semi-Stream CACHEJOIN for Near-Real-Time Data Warehousing

M. Asif Naeem, Auckland University of Technology, Auckland, New Zealand

 <https://orcid.org/0000-0001-6785-7875>

Erum Mehmood, School of Science and Technology, University of Management and Technology, Lahore, Pakistan

M. G. Abbas Malik, Universal College of Learning, Palmerston North, New Zealand

Noreen Jamil, National University FAST, Islamabad, Pakistan

## ABSTRACT

Streaming data join is a critical process in the field of near-real-time data warehousing. For this purpose, an adaptive semi-stream join algorithm called CACHEJOIN (Cache Join) focusing non-uniform stream data is provided in the literature. However, this algorithm cannot exploit the memory and CPU resources optimally and consequently it leaves its service rate suboptimal due to sequential execution of both of its phases, called stream-probing (SP) phase and disk-probing (DP) phase. By integrating the advantages of CACHEJOIN, this article presents two modifications for it. The first is called P-CACHEJOIN (Parallel Cache Join) that enables the parallel processing of two phases in CACHEJOIN. This increases number of joined stream records and therefore improves throughput considerably. The second is called OP-CACHEJOIN (Optimized Parallel Cache Join) that implements a parallel loading of stored data into memory while the DP phase is executing. This research presents the performance analysis of both of the approaches defined within the paper existing CACHEJOIN empirically using synthetic skewed dataset.

## KEYWORDS

Near-Real-Time Data Warehousing, Semi-Stream Join, Service Rate Optimization

## INTRODUCTION

In today's world, the real-time data availability for well-timed and well-informed decisions has become decisive for successful businesses, while data sizes are growing exponentially. Significance of real-time business data devalues, as it gets older. At the same time, the traditional working hours for global enterprises are not germane as they continue to serve customers around the globe and around the clock every day (Golfarelli & Rizzi, 2009; Vassiliadis, 2009; Thomsen & Pedersen, 2005). For uninterrupted global services, continuous real-time data availability for in time business decisions and actions is crucial and indispensable. Traditional offline data-refresh at data warehouses (DWHs) via ETL (Extract-Transform-Load) processes in batch windows (Kimball & Caserta, 2011) are not endurable in this scenario. Therefore, near-real-time data warehousing (NRT-DWH) is an evolving research area and plays a prominent role in supporting cutting-edge and contemporary business strategies and social requirements of the modern era. The modern warehousing techniques are transforming traditional warehouse from a static data repository into an active business entity. This

DOI: 10.4018/JDM.2020010102

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

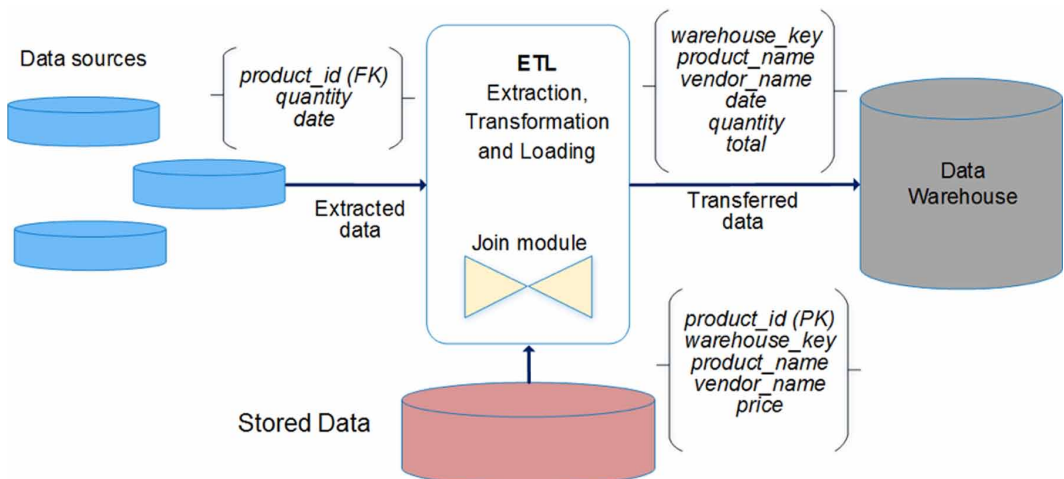
helps to fulfill the contemporary business needs ranging from informing the different stakeholders about latest updates to effective, timely and accurate business decisions.

According to the demand of DWH industry, there is a need to develop an efficient algorithm that performs join operation for bursty and fast streaming data. In NRT-DWH, relational data generated by different data sources needs to reflect in the DWH with a minimal possible delay. Because data is coming from numerous data sources within the organization, it requires significant cleansing and transformation before loading it into the DWH using SQL. Thus, the powerful SQL features can be used to gain consistency and ACID compatibility for join query from the relational schema (Irshad et al., 2019). ETL processes are used for this purpose (Kimball & Caserta, 2011; Bornea et al., 2011). Transformation of extracted data (user sales data) from numerous sources is a crucial phase in ETL processes. In this phase, a stream of new extracted data is joined with a stored data before loading this into the DWH, as shown in Figure 1. Typically, a foreign key from the stream data is joined with the primary key in the master data (Naeem et al., 2012a; Mokbel et al., 2004; Dittrich et al., 2002). Since the join is between the stream data and the stored data therefore, it is called a semi-stream join.

The problem of joining a streaming data with a stored data was first introduced in (Neoklis Polyzotis, Skiadopoulos, Vassiliadis, Simitsis, & Frantzell, 2008) and as a solution a seminal algorithm called MESHJOIN (Mesh Join) was presented. Later, various optimizations in MESHJOIN have been proposed (Bornea et al., 2011; Naeem et al., 2012a; Naeem et al., 2010; Naeem et al., 2013; Du & Zou, 2013; Naeem et al., 2012b). Since the concept of long tail is very common in sales data (Kleinberg, 2002), CACHEJOIN (Naeem et al., 2012a), one of these algorithms, was particularly designed for irregular streams by caching the frequent records of stored data. However, it executes its two SP and DP phases sequentially. Because of the sequential execution, the stream records are waiting unnecessarily before being processed. Thus, the algorithm cannot achieve optimal performance. Parallel execution of the SP and DP phases of CACHEJOIN can significantly speed up the joining process. Further details about limitations of CACHEJOIN are presented later in the paper.

In this paper we propose two modifications in the CACHEJOIN algorithm. First is called P-CACHEJOIN (Parallel Cache Join) (Mehmood & Naeem, 2017)<sup>1</sup> that deals the problem of sequential execution of two phases of CACHEJOIN algorithm. This proposed approach reduces the unnecessary waiting time for the stream records. Second is called OP-CACHEJOIN (Optimized Parallel Cache

Figure 1. Illustration of the join during the transformation phase of ETL



16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/article/optimizing-semi-stream-cachejoin-for-near-real-time-data-warehousing/245298](http://www.igi-global.com/article/optimizing-semi-stream-cachejoin-for-near-real-time-data-warehousing/245298)

## Related Content

---

### Database Replication Protocols

Francesc D. Muñoz-Escoí, Luis Irún-Brizand Hendrik Decker (2005). *Encyclopedia of Database Technologies and Applications* (pp. 153-157).

[www.irma-international.org/chapter/database-replication-protocols/11138](http://www.irma-international.org/chapter/database-replication-protocols/11138)

### Privacy-Preserving Data Mining: Development and Directions

Bhavani Thuraisingham (2005). *Journal of Database Management* (pp. 75-87).

[www.irma-international.org/article/privacy-preserving-data-mining/3328](http://www.irma-international.org/article/privacy-preserving-data-mining/3328)

### Raster Databases

Peter Baumann (2005). *Encyclopedia of Database Technologies and Applications* (pp. 517-523).

[www.irma-international.org/chapter/raster-databases/11198](http://www.irma-international.org/chapter/raster-databases/11198)

### Multi-Fuzzy-Objective Graph Pattern Matching with Big Graph Data

Lei Li, Fang Zhangand Guanfeng Liu (2019). *Journal of Database Management* (pp. 24-40).

[www.irma-international.org/article/multi-fuzzy-objective-graph-pattern-matching-with-big-graph-data/241830](http://www.irma-international.org/article/multi-fuzzy-objective-graph-pattern-matching-with-big-graph-data/241830)

### Maintenance of Association Rules Using Pre-Large Itemsets

Tzung-Pei Hongand Ching-Yao Wang (2007). *Intelligent Databases: Technologies and Applications* (pp. 44-60).

[www.irma-international.org/chapter/maintenance-association-rules-using-pre/24229](http://www.irma-international.org/chapter/maintenance-association-rules-using-pre/24229)