1. INTRODUCTION

Near-real-time data warehousing exploits the concepts of data freshness in traditional static data repositories in order to meet the required decision support capabilities. The tools and techniques for promoting these concepts are rapidly evolving (Thomsen & Pedersen, 2009; Golfarelli & Rizzi, 2009a, 2009b; Vassiliadis, 2009). Most data warehouses have already switched from a full refresh (Gupta & Mumick, 1999; Zhang & Rundensteiner, 2002; Zhuge, Garcia-Molina, Hammer, & Widom, 1995) to an incremental refresh policy (Labio & Garcia-Molina, 1996; Labio, Wiener, Garcia-Molina, & Gorelik, 2000; Labio, Yang, Cui, Garcia-Molina, & Widom, 2000). Furthermore, the batch-oriented, incremental refresh approach is moving towards a continuous, incremental refresh approach (Thiele, Fischer, & Lehner, 2007; Karakasidis, Vassiliadis, & Pitora, 2005; Nguyen, 2003).

DOI: 10.4018/jdwm.2011100102
With regards to terminology, data warehousing approaches that follow such a best-effort data freshness approach have various names. Frequently used terms are zero-latency, active, real-time or near-real-time data warehouses. The term near-real-time is the most descriptive in a context where there could be confusion with real-time control systems, but for the sake of brevity, we will mostly use the term real-time in this paper where no such confusion is possible.

One important research area in the field of data warehousing is data transformation, since the updates coming from the data sources are often not in the format required for the data warehouse. For real-time data warehousing a continuous transformation from a source to target format is required, so the task becomes more challenging.

In the ETL (Extract-Transform-Load) layer, a number of transformations are performed such as the detection of duplicate tuples, identification of newly inserted tuples, and the enriching of updates with values from the master data. Enrichment in particular can often be expressed as a join between the update stream and the master data (Naeem, Dobbie, & Weber, 2008). One important example of enrichment is a key transformation. Normally the key used in the data source is different from that in the data warehouse and therefore needs to be replaced. This transformation can be obtained by implementing a join operation between the update tuples and a lookup table. The lookup table contains the mapping between the source keys and the warehouse keys.

A novel stream-based equijoin algorithm, MESHJOIN (Polyzotis, Skiadopoulos, Vassiliadis, Simitsis, & Frantzell, 2007; Neoklis Polyzotis, Skiadopoulos, Vassiliadis, Simitsis, & Frantzell, 2008) is in principle a hash join, where the stream serves as the build input and the disk-based relation serves as the probe input. The main contribution is a staggered execution of the hash table build and an optimization of the disk buffer for the disk-based relation.
Related Content

Recent Developments on Security and Reliability in Large-Scale Data Processing with MapReduce

Robust Statistical Methods for Rapid Data Labelling
Jamie Godwin and Peter Matthews (2014). *Data Mining and Analysis in the Engineering Field* (pp. 107-141).
[www.igi-global.com/chapter/robust-statistical-methods-for-rapid-data-labelling/109979?camid=4v1a](www.igi-global.com/chapter/robust-statistical-methods-for-rapid-data-labelling/109979?camid=4v1a)
Improving Similarity Search in Time Series Using Wavelets
[www.igi-global.com/article/improving-similarity-search-time-series/1766?camid=4v1a](www.igi-global.com/article/improving-similarity-search-time-series/1766?camid=4v1a)

Online Clustering and Outlier Detection
[www.igi-global.com/chapter/online-clustering-outlier-detection/73438?camid=4v1a](www.igi-global.com/chapter/online-clustering-outlier-detection/73438?camid=4v1a)