TInTo: A Tool for View-Based Analysis of Stock Market Data Streams

Andreas Behrend, Christian Dorau, and Rainer Manthey

University of Bonn, Institute of Computer Science III
Roemerstr. 164, D-53117 Bonn, Germany
{behrend,dorau,manthey}@cs.uni-bonn.de

Abstract: TinTO is an experimental system aiming at demonstrating the usefulness and feasibility of incrementally evaluated SQL queries for analyzing a wide spectrum of data streams. As application area we have chosen the technical analysis of stock market data, mainly because this kind of application exhibits sufficiently many of those characteristics for which relational query technology can be reasonably considered in a stream context. TinTO is a technical investor tool for computing so-called technical indicators, numerical values calculated from a certain kind of stock market data, characterizing the development of stock prices over a given time period. Update propagation is employed for the incremental recomputation of indicator views defined over a stream of continuously changing price data.

1 Technical Analysis of Stock Market Data

Technical analysis (TA) is concerned with the prediction of future developments of stock market prices. TA uses so-called technical indicators, numerical values derived from the past development of prices of a certain stock. In principle, indicators are functions applied to the price history of a certain stock and a point in time. A technical analyst is usually interested in the change of indicator values over a certain time period in order to automatically derive buy or sell signals for stocks. The stream of time-stamped price data is an ordered append-only stream provided by a stock market such as the New York stock exchange. In general, indicator computation is based on a division of a stocks price history into consecutive time intervals t_i of equal length. Interval length is user-defined and usually ranges from 1 second to as long as one year. For each time interval, the mean average is determined representing the typical price TP of stock s with respect to t_i. A technical indicator definition is based on these values and an additional user-defined parameter n specifying the number of consecutive time intervals under consideration. As an example, consider the commodity channel index (CCI) which is an overbought/oversold indicator, whose value typically oscillates around zero:

$$\mathrm{CCI}_n(s,t_i) := \frac{\mathrm{TP}(s,t_i) - \frac{\sum_{l=0}^{n-1} \mathrm{TP}(s,t_{i-l})}{n}}{\frac{0.015}{n} \cdot \sum_{k=0}^{n-1} \left|\mathrm{TP}(s,t_i) - \frac{\sum_{l=0}^{n-1} \mathrm{TP}(s,t_{i-l-k})}{n}\right|}$$

Figure 1: Main window of TinTO

Readings above +100 imply an overbought condition, while readings below -100 imply an oversold situation. When the CCI moves above -100, this is interpreted as the beginning of a positive price trend and, thus, may serve as a buy signal. The position should be closed when the CCI reaches +100 as it enters an overbought condition. CCI values can be analogously used as short selling signals.

Indicator definitions may be based on other ones, forming a kind of indicator hierarchy. In this way, it is possible to combine the positive effects of different indicator types. To which extent technical indicators are meaningful for predicting future price developments, however, remains quite questionable and no statistical significant forecast power has been proved so far. Nevertheless, technical indicators represent a quite general way of analyzing a numerical data stream and can be seen as continuous queries of considerable complexity. As these (potentially recursive) queries are general aggregation functions, they are quite representative for a broad class of stream analysis problems. Although there are various commercial implementations of technical indicators, technical analysts are often interested in building their own indicator system in order to develop a personalized and unique trading strategy. To this end, the appropriateness of new parameter values is investigated and new indicators are invented in order to improve forecast power. A possible way of achieving a flexible and extendible trading system is to implement indicators using a declarative language like SQL.

2 The TinTo System

The acronym TInTo is used as an abbreviation for Technical Investor Tool indicating its task as automatic trading system based on indicator signals. Initially, TInTo has been a Visual Basic (VBA) application based on MS Access only, but is currently being reprogrammed as a web-based application using Java and Oracle. TInTo manages a user-defined portfolio of indices, commodities, and stocks whose historical as well as intraday quotations are provided by `http://finance.yahoo.com` for stock markets worldwide. As a frontend it uses the shareware visualizer ChartDirector [Cha06], a tool supplying a VBA library of well-established methods for drawing financial charts (cf. Figure 1).

TInTo provides a simple SQL view editor for specifying arbitrary technical indicators as predefined SQL queries (i.e., as views), evaluated directly over the underlying database containing timestamped price data. The advantage of view-based indicators is that the underlying definition can be easily recovered and modified while new indicators can be simply defined in form of view hierarchies. In addition, the application of SQL views allows the efficient computation of indicator values directly within the database where the portfolio and price data is usually stored, thus avoiding the well-known impedance mismatch. At present we experiment with some 30 standard indicators as proposed in [BL92] and four new ones where we tried to combine the positive effects of various indicators of different type. For testing the quality of trading signals, an evaluation tool is provided by TInTo which calculates the capital gains and losses achieved when a user trades a chosen commodity according to the signals generated by a selected indicator. To this end, a user must specify a time period over which the corresponding signals are calculated and percentaged gains and losses are accumulated.

Even though a considerable degree of analysis is reachable this way, hardly any streaming is involved yet. However, there are so-called intraday trading strategies which need to access a high frequency stream. The crucial step towards proper stream management in TinTO consisted in the addition of a simple VBA script automatically downloading a record of characteristic values per stock in the portfolio at regular intervals and appending the downloaded data to those already present in the database. This component generates a data stream with a frequency of up to one second pulled from Yahoo on demand.

3 Incremental Evaluation of Continuous Queries

The main goal of TinTo is to provide a performance analysis of incrementally evaluated SQL queries over data streams. In data stream research, it is widely believed that conventional relational database systems are not well-suited for dynamically processing continuous queries. We believe, however, that even conventional SQL queries can be efficiently employed for analyzing a wide spectrum of data streams. In particular, we think that the application of incremental update propagation considerably improves the efficiency of computing answers to continuous queries. Update propagation is not a new research topic but has been intensively studied for many years mainly in the context of integrity checking and materialized views maintenance [GM99]. The key idea is to transform each SQL view already at schema design time into a so-called *delta view*, a specialized version of the view referring to changes in the underlying tables, only. The original view definitions are employed only once for materializing their initial answers while the specialized versions are used afterwards for continuously updating the materialized results. Under the assumption that a great portion of the materialized view content remains unchanged, the application of delta views may considerably enhance the efficiency of view maintenance. We adopted this idea for the TInTo system, using delta-view techniques for a synchronized update of indicator values. To this end, update statements are applied instead of the original indicator views for incrementally maintaining the materialized indicator values. In principle, these update statements can be automatically compiled from the original views. However, we do not have a full-fledged delta compiler for arbitrary SQL views yet, therefore we are performing our experiments with hand-compiled delta views for the time being. In [ABS08], however, the author et al. describe a way of deriving such specialized update statements systematically.

All indicator queries are based on a sliding window defined by the time span for which the user wants to observe indicator values. In TInTo, a user determines the window size by setting the time range attribute which may take values from 1 day to 20 years. As an example, consider CCI values for a period of one year:

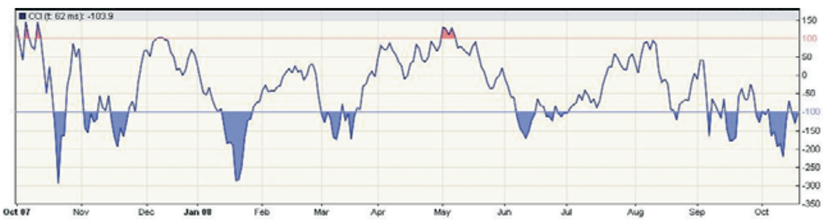

Figure 2: Continuous recomputation of CCI values

As soon as a new current time interval t_c^{new} is considered, the oldest one and its corresponding indicator values are removed from the list of tuples to be displayed, and the new entry $CCI(t_c^{new}) = -103,9$ is calculated for t_c^{new}. The time needed for the incremental computation of $CCI(t_c^{new})$ is `62 ms` as indicated in the upper left corner of Figure 2. A corresponding non-incremental computation takes `336 ms` on average. At the demonstration site, we will show this significant performance gain for other window sizes and other types of indicators (e.g. including recursive ones). These measurements show that this kind of performance gain fully scales with the size of the sliding window. The performance results provide first evidence that incremental evaluation of SQL views represents a suitable approach for analyzing a wide spectrum of data streams.

References

[ABS08] R. Manthey A. Behrend, C. Dorau and G. Schüller. Incremental View-Based Analysis of Stock Market Data Streams. In *IDEAS*, pages 269–275, 2008.

[BL92] C. Le Beau and D. Lucas. *Technical Traders Guide to Computer Analysis of the Futures Markets*. Irwin Professional (USA), 1992.

[Cha06] Chart Director, 2006. [Online; 09.10.2006].

[GM99] Ashish Gupta and Inderpal Singh Mumick. *Materialized Views: Techniques, Implementations, and Applications*. The MIT Press, 1999.