

Federal Market Information Technology in the Post Flash Crash Era: Roles for Supercomputing*

E. Wes Bethel^{† ‡}
Visualization Group,
CRD, LBNL,
Berkeley, CA, USA
ewbethel@lbl.gov

David Leinweber
Center for Innovative
Financial Technology,
CRD, LBNL
Berkeley, CA, USA
dleinweber@lbl.gov

Oliver Rübel
Visualization Group,
CRD, LBNL,
Berkeley, CA, USA
oruebel@lbl.gov

Kesheng Wu
Scientific Data
Management Group
CRD, LBNL
Berkeley, CA, USA
kwu@lbl.gov

ABSTRACT

This paper describes collaborative work between active traders, regulators, economists, and supercomputing researchers to replicate and extend investigations of the Flash Crash and other market anomalies in a National Laboratory HPC environment. Our work suggests that supercomputing tools and methods will be valuable to market regulators in achieving the goal of market safety, stability, and security.

Currently the key mechanism for preventing catastrophic market action are “circuit breakers.” We believe a more graduated approach, similar to the “yellow light” approach in motorsports to slow down traffic, might be a better way to achieve the same goal. To enable this objective, we study a number of indicators that could foresee hazards in market conditions and explore options to confirm such predictions. Our tests confirm that Volume Synchronized Probability of Informed Trading (VPIN) and a version of volume Herfindahl-Hirschman Index (HHI) for measuring mar-

ket fragmentation can indeed give strong signals ahead of the Flash Crash event on May 6 2010. This is a preliminary step toward a full-fledged early-warning system for unusual market conditions.

Categories and Subject Descriptors

J.0 [Computer Applications]: General

General Terms

Design, Performance

Keywords

Flash crash, liquidity, flow toxicity, market microstructure, probability of informed trading, VPIN, circuit breakers, market fragmentation. (JEL codes: C02, D52, D53, G14.)

*This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

[†]Authors are listed in alphabetical order.

[‡]CRD: Computational Research Division
LBNL: Lawrence Berkeley National Laboratory

1. INTRODUCTION

After the dramatic Flash Crash of May 6, 2010, it took more than four months in order for the SEC/CFTC to issue a full report on the event [2,3]. Such a long duration of time was required because the government currently relies on a mélange of legacy systems. The SEC and CFTC clearly realized the limitations, and have called for comments in a 203-page discussion of the proposed next-generation system: CATS (Consolidated Audit Trail System) [18]. In computer science, visionary scientists like Jim Gray have developed technologies and tools to foster a brand new approach for scientific discoveries termed “data-intensive science [10]”. This paper reports our initial attempt to adapt some key techniques to address the information technology needs of financial regulatory agencies.

A basic tool in regulating the financial market is the “circuit breaker” that stops trading. However, many traders and academicians have compared this approach of on/off circuit breakers to “applying the rules of the road to aircraft.” As different markets and venues become more interdependent, sudden halts in one market segment can ripple into others [13]. One key observation is that brown-outs are preferable to black-outs. After the Flash Crash of 2010, new circuit breakers were instituted that stop the trading of individual stocks if their price variations exceed a prescribed threshold. However, this approach is insufficient for achieving safety and stability. We propose to detect and predict hazardous conditions in real-time. If successful, this would

allow the regulatory agencies to implement a “yellow light” approach, to slow down, rather than stop markets.

In this work, we seek to explore three aspects of implementing these early warning indicators: i) finding indicators that can provide early warnings, ii) better understanding of their computational requirements, and iii) implementing an interactive exploratory system for validation of warning indicators and to allow expert verification in cases requiring extraordinary actions.

Based on recommendations from large traders, regulators, and academicians, we have implemented two sets of indicators from institutional traders that have been shown to have “early warning” properties preceding the Flash Crash. They are the Volume Synchronized Probability of Informed Trading [7] (VPIN) and a variant of Herfindahl-Hirschman Index (HHI) of market fragmentation [9, 11]. We describe how to organize data to efficiently compute these indicators and how to parallelize the computational tasks on a high performance computer (HPC) system. Because the automated indicators may need to be verified if a critical action is to be taken, we also explore the option of interactively verifying warnings with comparisons to historical data by using a high-performance search and visualization tool. One major operational challenge in the future will be to perform all these tasks – including computation and verification of market warnings – in real-time.

2. BACKGROUND

2.1 Levels of Financial Market Data

In this section, we briefly describe the types of data used in our work. Literature on market microstructure often refer to six levels of market data [16], due to space limitation, we mention the first three levels.

2.1.1 Level 1 – Transactions

The most basic form of market data is a time series of trades or transactions, typically including price, volume, time and trading venue for each trade. Many financial time series, such as the Dow-Jones Industrial Average, are weighted averages of stock trades.

2.1.2 Level 2 – BBO Quotes

Level 2 data consists of all transactions from Level 1, plus the “inside quotes,” i.e., the best bid and offer price for each security in the combined markets included in the data. The best known example of Level 2 data is the Trades And Quotes (TAQ) data originally developed by the New York Stock Exchange [15]. It is used extensively by the research community. Current version of TAQ data includes Level 2 data from many market fragments including NYSE, NASDAQ, ARCA, and BATS. This type of data contains considerably more information than Level 1 data, but at about a few GB per day it is also quite convenient for analysis and is used in our current work. A sample using TAQ data to describe the Flash Crash event on May 6, 2010 is shown in Figure 1.

2.1.3 Level 3 – Limit Order Book

In a typical exchange, there is an electronic Limit Order Book (LOB) that contains all current waiting quotes. The Best Bid/Offer (BBO) quotes included in Level 2 are the quotes that are closest to each other for a stock. There

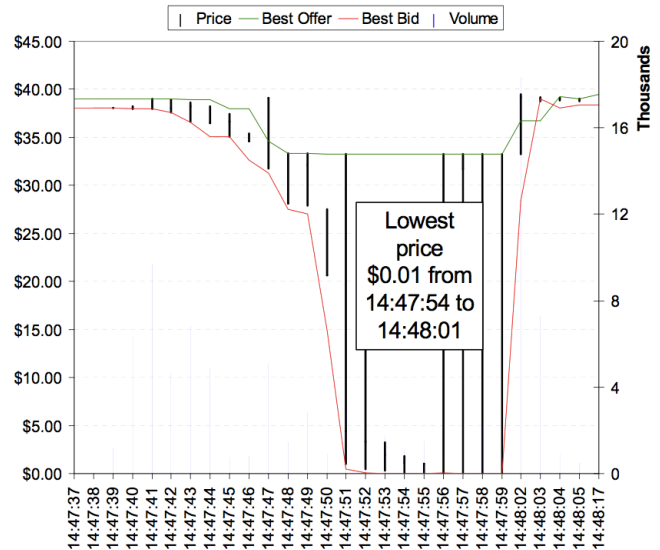


Figure 1: Using Level 2 data shows the bizarre behavior in the trades and quotes for Accenture (ACN), one of the stocks most extremely impacted during the Flash Crash. (Source: May 18 CFTC/SEC report, pg 35)

are many limit orders in the LOB with prices away from the BBO. With decimal penny pricing replacing the old 1/8 and 1/16, these quotes have become more important. With ever faster electronic market access, they could change very rapidly, in milli- to micro-seconds. A typical LOB data set is a snapshot in time, showing prices, and total sizes at each price. In some versions, the breakdown of the many orders comprising the total size at a price is also included.

2.2 Market Indicators for Early Warning of Anomalies

Numerous indicators have been devised to measure aspects of the financial market. We start by examining a few indicators that are known to provide early warning signals for the Flash Crash of May 6, 2010. They are the Volume-Synchronized Probability of Informed Trading (VPIN) [7] and the Herfindahl-Hirschman Index (HHI) [9, 11]. Because these indicators revealed unusual behavior on May 6, 2010, they might also detect other unusual activities.

VPIN measures the balance between buy and sell activities [7]. An earlier version of this indicator is called Probability of Informed Trading (PIN) [6]. The key change in VPIN is to use bins with the same trading volume instead of bins with the same time span. The VPIN authors refer to this as measuring the buy-sell imbalance in volume-time instead of clock-time. Furthermore, instead of using the relative imbalance value directly, which can be different for different commodities, the authors normalized them using the function Φ that defines the Cumulative Distribution Function of a normal distribution. Because of this normalization, a single threshold, $T = 0.9$, can be used for many different stocks. With suitable parameters, the authors have shown that the VPIN reaches 0.9 more than an hour before the Flash Crash on May 6, 2010. This is the strongest early warning signal known to us at this time. To a large extent,

our computation of VPIN replicates the one used by Easley et al. [6]. In fact, the developers of VPIN have shared a Python implementation of their program with us and our C++ implementation reproduces exactly the same values on the same input values. A key difference is that we compute VPIN values of individual stocks while the earlier work computes VPIN values on SP500 futures.

Another indicator producing a clear early warning signal for the Flash Crash of 2010 was a market fragmentation measure based on HHI [14]. The particular version used in this work is called the Volume Herfindahl Index, but many other variations exist in the literature. Because we only use the Volume Herfindahl Index in this work, we simply refer to it as HHI in later discussions. During a given time window, say five minutes, the fractions of trade volumes executed by different stock exchanges can be computed. HHI is the sum of squares of these fractions [9, 11]. Variations of HHI are widely used to measure the concentration of industrial production and other economic power [5, 20].

In [14], Madhavan computes a single HHI value for an entire day. In an attempt to use HHI as an early warning indicator, Madhavan suggested that we break each day into small intervals. Here, we choose to use 5-minute bins. Furthermore, to detect “abnormal” values, we define a reference window of twelve bins that covers the hour preceding the current bin. We use the bins in the reference window to compute a mean and a standard deviation. We declare an HHI to be “abnormal” if it is more than x times the standard deviation away from the mean. Note that an x value of 1.645 is equivalent to the choice of 0.9 as the threshold for VPIN.

2.3 Data Management

In order to perform the above mentioned computations effectively, the required data must be in the appropriate format. For example, the widely available TAQ (Trades And Quotes) data [15] is available typically on a CD or DVD with extraction program that runs only on MS Windows platforms. Since most of the HPC systems run Linux operating systems, TAQ data requires a transformation step before it can be used on HPC platforms. Many other collections of data have a similar limitation.

Even if the computing platform is based on MS Windows, the data extraction program, such as the one provided with the TAQ data distribution, produces Comma-Separated Values (CSV), i.e., an ASCII representation of the values. This representation typically requires more bytes than the corresponding binary representation and requires significantly more time to read into memory. Because the data records in ASCII require different numbers of bytes, it is more difficult to skip unwanted bytes to directly extract a specific data record. Other data distributions, e.g., by Nanex, do not require an intermediate conversion to ASCII. However, many of them still require the user to go through each data record, without obvious means for skipping unwanted records.

One way to provide high-performance data access is to store the data records in a commercial database management system (DBMS). Some DBMS have extensive support for operations on financial data series. However, to achieve higher performance and to have more control over the analysis operations, we have taken the approach used by many scientific applications – using a high-level data format library,

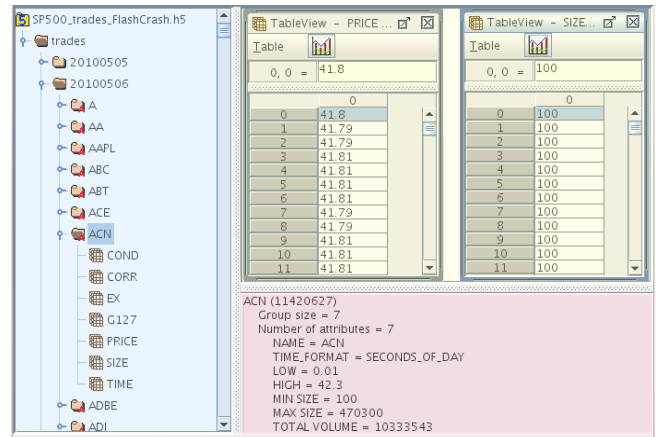


Figure 2: HDF5 data-layout used for storing trades data showing: i) the organization of the data via HDF5 groups and datasets (blue), ii) example meta-data for ACN, 05/06/2010 (red), and iii) spreadsheet-view of PRICE and SIZE data array for ACN, 05/06/2010 (yellow).

more specifically the Hierarchical Data Format version 5, or HDF5 [8].

3. CASE STUDY

The goal of this case study is to evaluate how high-performance computing can support financial data analysis and, in particular, the development and implementation of early warning systems for detection and analysis of market anomalies. Development and evaluation of reliable indicators for market anomalies requires thorough analysis of the effectiveness of such indicators on large amounts of historic data. We need to be able to: i) store and process large amounts of data, ii) efficiently compute market indicators, and iii) quickly extract and analyze portions of data during which abnormal market behavior is indicated.

3.1 File Format

Enabling efficient analysis of large amounts of data fundamentally relies on effective data organization and storage to optimize I/O performance. In this work we adopt HDF5 [8] – a state-of-the-art, open, scientific data format – for storing financial data. Figure 2 illustrates the HDF5 data-layout we are using for storing TAQ data. We organize the data into groups based on the data type (*trades*, *quotes*), date, and stock symbol. Each complete group (e.g. *trades/20100506/ACN*) then contains a set of 1D HDF5 datasets of varying types (e.g., *PRICE* stored as floats, or *SIZE* stored as integers). Additional information about the data, like the time format, and simple statistics are stored as HDF5 attributes associated with the corresponding datasets and groups. SZIP compression can further significantly reduce file size, by a factor of 5-7 in the case of TAQ data, while enabling fast decompression and, hence, data access. Table 1 compares the storage requirements of an example TAQ dataset in different formats.

Using HDF5 for storing financial data has many advantages. HDF5 is portable, easy to use, efficient with respect to storage and I/O performance, supports compression and

	CSV	CSV (zip)	HDF5	HDF5 (SZIP)	Index
Trades	2,769	215	1,326	472	1,803
Quotes	38,566	3,058	28,844	5,377	24,784

Table 1: File sizes in mega-byte (MB) for example TAQ data using different file formats. The datasets contain three days worth of trades and quotes for S&P 500 symbols.

parallel I/O, and provides additional tools for browsing, validation and profiling (e.g., HDFView). HDF5 is optimized for large data files and enables us to store months to years worth of financial data in a single file while maintaining easy data access. Furthermore, HDF5 enables the use of advanced HPC index and search software [17]. In our case, we use the FastQuery software to index and search the data [4].

3.2 Computing Market Indicators

After organizing the data in an efficient way, the next question to answer is whether HPC resources can effectively compute market indicators. We use the computations of VPIN and HHI as examples, and treat the computation of VPIN and HHI on each stock or fund as a separate computational task. Because these tasks don't require any coordination among them, they can achieve good speedup as we show later. A key limitation to achieving the perfect speedup is the load imbalance. To minimize this imbalance, we dynamically assign work to each process using the manager-worker approach. To further increase scalability of the computations on long-term historic data, one could further subdivide the calculations into a series of time-intervals. In this section, we also present some evidence that HHI and VPIN produce strong signals before and during the Flash Crash of 2010.

We start our discussion on the computation of VPIN and HHI by describing the data used. We use Level 1 data, i.e., trades, for computing VPIN and HHI. The test data is divided into two sets. The first set of trades covers the time period of April and May, 2010, containing 45 trading days. This set contains all trades of SP500 stocks. The total number of records is about 640 million and the total size is about 25 GB as CSV files and 4.4 GB as HDF5 files.

The second set contains trades of 25 ETFs with the largest trading volumes. The time period varies from 3 years (2008, 2009, and 2010) to 10 years (2001 – 2010). The total number of records is about 2.7 billion, and the size is 108 GB in CSV and 17 GB in HDF5. Clearly, there is a size advantage for using HDF5 files.

It is faster to use data in HDF5 files for computation as well. For example, on a subset of data from May 2010, using HDF5 it only took 0.4 seconds to compute VPIN for Accenture (ACN) stock. However, it took 142 seconds using the corresponding CSV files. Using HDF5 files speeds up the VPIN computation by a factor of 355. The key difference between using HDF5 and CSV is that using HDF5 files, combined with efficient indexing, one can quickly locate the desired data records, while using CSV files, one has to read through each data record to locate the desired ACN records.

We compute VPIN and HHI for each stock or fund separately, in order to raise the “yellow flag” on each of them independently. We realize that this is not exactly how the orig-

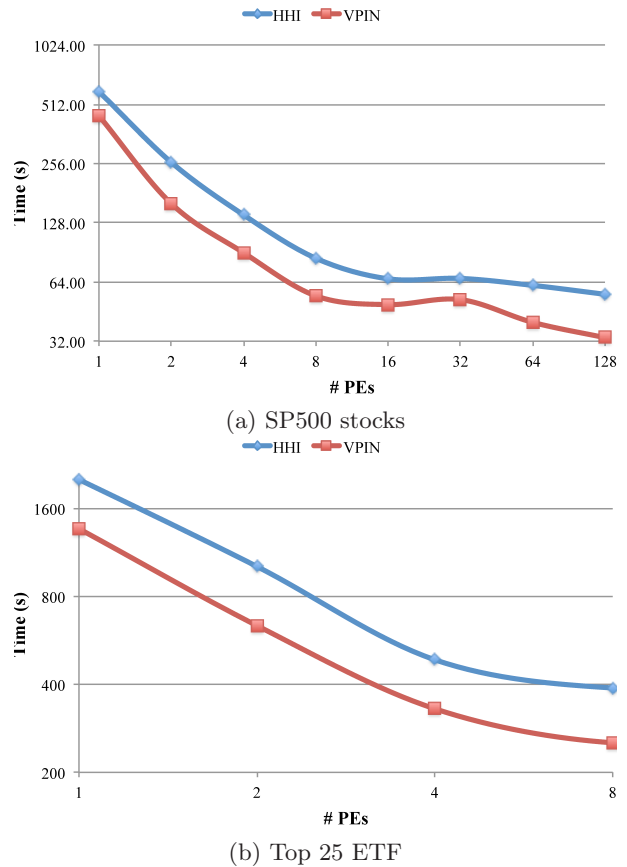


Figure 4: Time (seconds) needed to compute HHI and VPIN using different numbers of processing elements (PEs). The computation is broken up into 500 independent tasks for SP500 data and 25 tasks for the 25 ETFs. The maximum speedup achieved is about 11 for computing HHI and 13 for computing VPIN on SP500 data and about 5 for computing both HHI and VPIN on the 25 ETFs.

inal authors of VPIN and HHI intended to use them [7, 14]. However, as we show in Figure 3, there are strong evidences that VPIN and HHI can indeed provide early warning for the Flash Crash of May 6, 2010.

In Figure 3 we show the values of VPIN and HHI for four individual stocks: Accenture (ACN), CenterPoint Energy (CNP), Hewlett-Packard (HPQ), and Apple (AAPL). The first two are the well known examples where the prices dropped to one penny per share during the Flash Crash, HPQ is one of the Dow-Jones stocks significantly affected by the Flash Crash, and AAPL stock has the unusual behavior of reaching to \$100,000 per share during the same time period.

The data shown in Figure 3 is from May 6, 2010. For each 5-minute time interval used to compute HHI, we also compute the minimum price and maximum price during the same time interval. In Figure 3, we see that the minimum price drops to one penny in a number of bins, the earliest of which is around 14:45. Not all the stock prices fell during the Flash Crash, in Figure 3 we see a small jump for HPQ and two sets of tremendous jumps for AAPL.

In Figure 3, the values of VPIN became quite high before

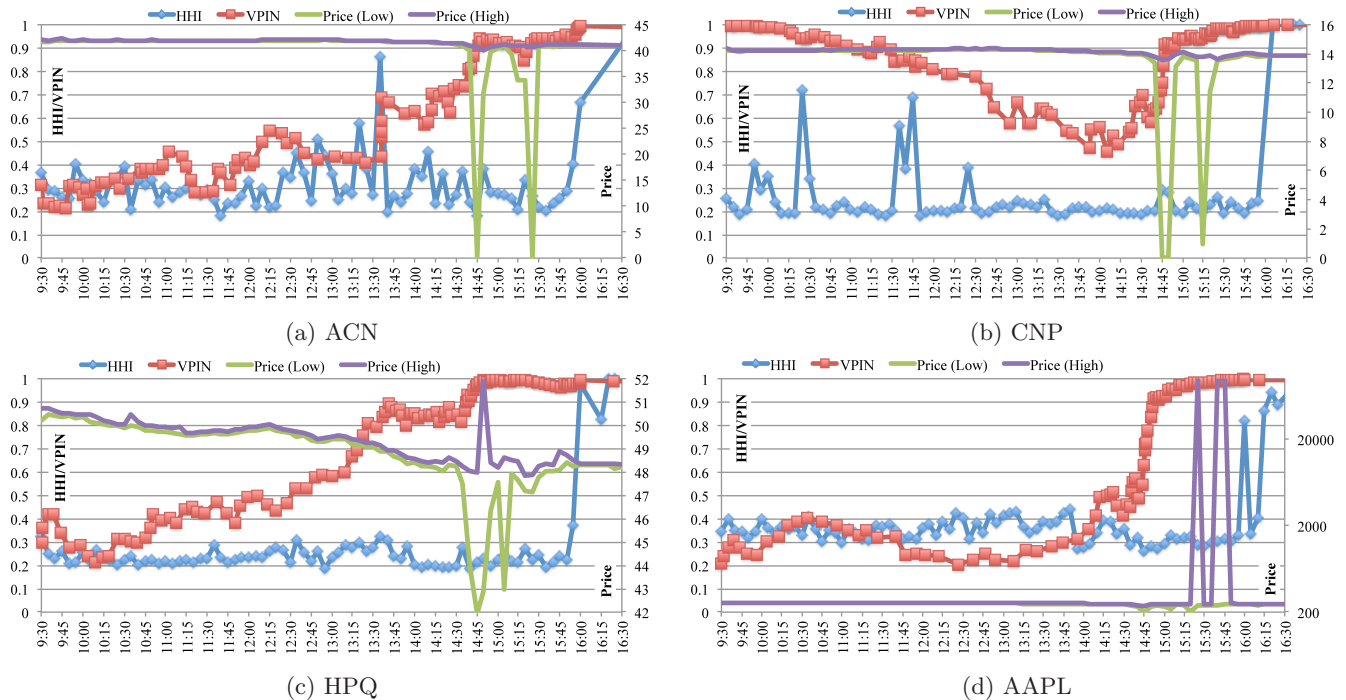


Figure 3: HHI (blue) and VPIN (red) values on May 6, 2010. The minimum and maximum prices shown are computed in each 5-minute bin. Each indicator shows extreme values before the irregular price changes.

the Flash Crash at 14:45. In the case of ACN, there was a sharp rise for both HHI and VPIN at 13:35¹. This was about 70 minutes before the Flash Crash.

For other stocks, either VPIN or HHI showed similar early warning before the Flash Crash. For example, for CNP, VPIN values were very high earlier in the day, for HPQ, VPIN also reached a high level around 13:45 (≈ 1 h before the Flash Crash), for AAPL, VPIN reaches a high level at 14:45 about half an hour before the unusual event at 15:15. Combined with the evidences provided by other authors [7, 14], we believe that VPIN and HHI are strong candidates for providing early warning signals of unusual activities.

Figure 4 shows how the time needed to compute HHI and VPIN varies with the number of processors used. The system used has 24 CPU cores on a compute node and each node has two network interface to other compute nodes and file systems². In Figure 4(a), we show the time to compute HHI and VPIN on SP500 stocks. In this case, the task of computing HHI or VPIN on each stock is given to a process that uses one CPU core on a node. These tasks are independent of each other in terms of computation, however, they need to access the same file system in order to read the necessary input data and write the final output. In Figure 4(b), we show the time to compute HHI and VPIN on the 25 ETFs. Again, we distributed the task of computing

HHI or VPIN for each ETF to a process. Because there are only 25 tasks, we used a maximum of 8 processing elements.

Overall, as the number of processes increase, the time needed goes down. However, due to load imbalance among the tasks and performance variance of the file system, time does not vary smoothly. For example, in Figure 4(a), as the number of PEs vary from 16 to 32, the total execution time appears to remain the same or slightly increase.

3.3 Query-driven Analysis and Data Exploration

Typical scientific applications require single (or few) queries to be evaluated on extremely large data. In contrast, screening of financial data requires evaluation of a large number of independent data queries (one query per *date/symbol* combination). Similarly, validation of market indicators fundamentally relies on the ability to quickly locate and extract data associated with large numbers of indicated warning periods, e.g, for HHI we find 298,956 potential warnings for S&P 500 stocks during April 2010.

To allow analysts to quickly define large sets of queries, we extend the FastQuery query language using symbolic queries. A symbolic query is a compact representation of a large number of queries using reserved keywords (here $\$DATE$ and $\$SYMBOL$) to represent data categories. The user can then select, from simple lists, the specific dates and symbols for which a symbolic query should be executed. The symbolic query is then automatically expanded into $\#dates * \#symbols$ queries. We use standard spreadsheet and statistics plots (Figure 6) for validation of queries and market-indicator warnings.

In the case of market indicators, large sets of warning events are created automatically by the screening process.

¹The particularly sharp rise in VPIN and HHI is possibly linked to a unusually large trade at 13:36:07. The volume of this single trade is 470,300 shares, which is almost 10% of the average daily volume for ACN. The impact of such a large trade on VPIN and HHI needs to be further examined.

²More information about the particular computer can be found at <http://www.nersc.gov/systems/hopper-cray-xe6/>.

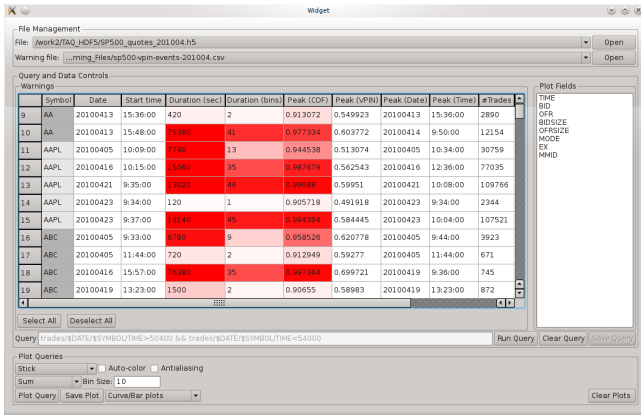


Figure 5: User interface for evaluating market indicator-based warnings.

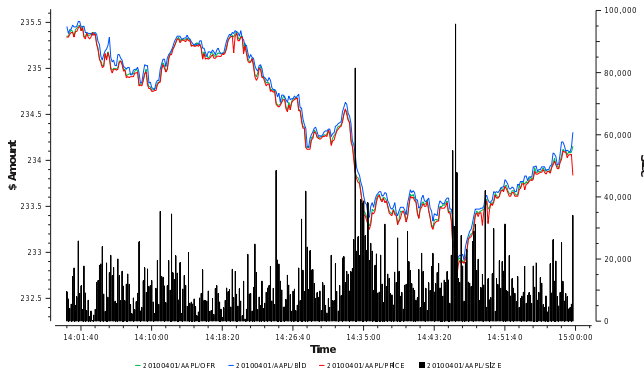


Figure 6: Plot created using our query tool showing the best OFR (green), BID (blue), PRICE (red), and total traded volume (SIZE)(black).

Each warning has an associated date, symbol, and time period. To extract the data associated with warning events, we automatically translate warnings to queries of the form:

$$\begin{aligned} TYPE/\$DATE/\$SYMBOL/TIME >= start\&\& \\ TYPE/\$DATE/\$SYMBOL/TIME <= stop, \end{aligned} \quad (1)$$

where *TYPE* identifies the trades/quotes group. We use a spreadsheet plot to allow analysts to quickly browse and select warnings of interest (Figure 5). The duration and peak warning value are color-coded to ease identification of long and high-risk warnings. Once a warning query has been executed, additional information –such as the number of trades/quotes executed during the warning period– are added to the table. To speed-up the analysis of false alarms, we plan to extend this concept with automatic methods for defining whether an anomaly occurred during a warning period.

In the following, we study the parallel performance of our query system, using the Hopper² system at NERSC. In the scaling experiment, we evaluate 8000 independent queries as defined in Eq. 1 on S&P 500 quotes data for April 2010. Each date/symbol group contains on average $\approx 274,151$ records, and 47 of the 8838 date/symbol groups contain 1.5 to 3.9 million records (Figure 7). This constitutes 74.5 GB of uncompressed HDF5 data (13.4 GB

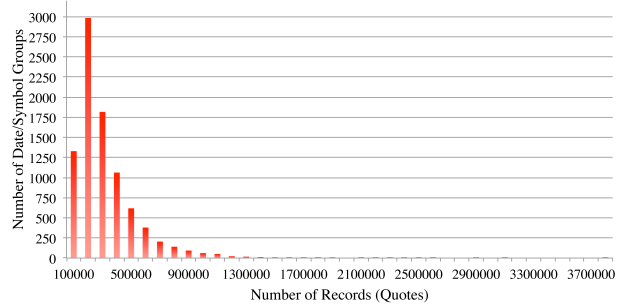


Figure 7: Number of data records (quotes) per date/symbol group for S&P 500 quotes data for April 2010.

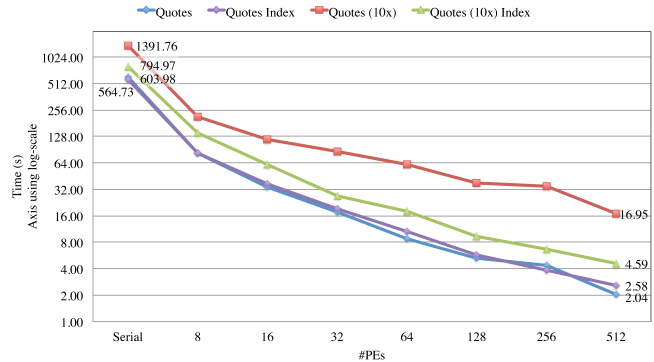


Figure 8: Time (seconds) needed to evaluate 8000 queries using different numbers of processors on i) S&P 500 quotes data for April 2010 (blue) and with indexing (lilac) and ii) the same dataset with 10 \times replication (red) and with indexing (green). Note the log-scale on the vertical time axis.

as HDF5 with SZIP compression and 100.4 GB as uncompressed csv) and 28.3 GB for all FastBit indices. To evaluate the expected performance on larger amounts of financial data, we replicated each HDF5 dataset 10 times, effectively increasing the number of quotes per date and symbol by a factor of 10. The replication also emulates storing ten month’s worth of quotes data grouped into periods of ten trading days. The replicated dataset constitutes 744.7 GB of uncompressed HDF5 data and 210.2 GB for all FastBit indices. Both quotes HDF5 files were created using default HDF5 settings and are stored on Hopper’s Lustre file system using a striping of 24 and a stripe-size of 1MB. We use a controller-worker-type setup implemented in MPI to parallelize the query process. The controller schedules the queries in batches of 10 queries as workers become available. Once all queries are completed, the controller acquires all hit-counts from the workers, while the file offsets (results) are stored on the workers to allow for efficient parallel analysis of the data associated with the queries. We repeated each experiment ten times and report the average wall-clock time elapsed to evaluate all queries, including all communication.

Figure 8 shows the results of this parallel query study. We observe good scalability in all cases and achieve two orders of magnitude speedup compared to the serial case. Using CSV data, it takes ≈ 3.5 hours in serial to evaluate all 8000

queries, even though we can compute all queries in a single pass, exploiting the fact that data records are sorted by date, symbol and time in the CSV file(s). Using HDF5 and indexing, we can perform the same analysis in parallel in less than 5 seconds on both datasets; a three orders of magnitude speed-up compared to using CSV. For the regular quotes data we do not observe a significant difference between the version with and without FastBit indexing. This behavior is expected, since the number of records required per query is low. For the dataset with 10× replication, we observe speedups of $\approx 2\text{-}4\times$ when using indexing. Interestingly, the time used for evaluating queries on the 10× replicated data does not increase by a factor of 10 compared to the regular quotes data. This is likely due to inefficiencies with respect to the HDF5 write: the default chunk-size may be too large for the small numbers of records per dataset, and/or the filesystem performance, which is sensitive to tunable parameters like the level of striping and stripe size, resulting in a larger I/O overhead for the smaller dataset. This type of variation in performance is well known: it has been shown that tuning of HDF5 and file system parameters can have a significant impact on I/O performance [12].

4. DISCUSSION

Data organization.

In this work, we demonstrated the benefit of using a more efficient data organization by using the HDF5 file format. For computing market indicators (Section 3.2), we see speedups of $\approx 355\times$ by using HDF5 compared to using CSV. This speedup results directly from more efficient data access methods. Organizing and storing financial data more efficiently has tremendous potential to improve the monitoring and reporting of financial markets.

Parallel computing.

Large-scale analyses of financial data –e.g., query-based analysis of historical data– require processing of large amounts of computationally independent data. The experiments in Sections 3.2 and 3.3 have shown that we can parallelize these types of operations effectively. The combination of efficient data organization and parallel computing enables us to evaluate large amounts of queries in seconds rather than hours. Such results, which stem from a combination of parallel computing, efficient data I/O and index/search, are consistent with other studies in forensic cybersecurity analysis [1, 19] and large-scale scientific data analysis [17], where processing time was reduced from hours or days to minutes or seconds. We are able to evaluate 8000 queries on a 74.5GB dataset in ≈ 2 seconds, an operation that takes more than 3.5 hours in serial using CSV data, a speedup of about 6,300×. The same operation on the larger 744.7GB dataset still requires less than 5 seconds.

HPC for early warning systems.

Based on our preliminary results, we believe that early warning systems can benefit substantially from HPC research, systems, and tools. Evaluation of the effectiveness of potential early warning indicators requires screening of large amounts of historical data. Ultimately we need to be able to compute market indicators in real time, requiring massively parallel algorithms and high-throughput data net-

works to analyze data from large numbers of stocks at once. At the same time, the real-time data needs to be stored efficiently for later analysis. For example, to enable regulators to judge the credibility of an indicated alarm, they need to be able to quickly locate and analyze similar events in large amounts of historical data. While our case study was limited in scope, focusing only on TAQ data, the results indicate that HPC methods can facilitate many of the tasks necessary for development, operation, and monitoring of a market monitoring and alarm system. For use scenarios involving larger amounts of financial data, we expect the computational demands to increase significantly making HPC methods indispensable.

Future Work.

In the current work, we have encountered inconsistent data from different sources. For example, different sources disagree on how many trades of AAPL at \$100,000 per share occurred on May 6, 2010. In the TAQ data, there are four records of trades at this price, two at 3:29:30 PM with a total of 895 shares, one at 3:44:51 PM with 695 shares, and one at 3:49:39 PM with 200 shares. In the Nanex data, there are also four trades recorded, though the volumes match those from TAQ, all of the records in Nanex data have the time stamp of 3:29:30 PM. The official SEC/CFTC report about the Flash Crash only mentioned two trades at 3:29:30 PM with a total volume of 895 shares. Clearly, such a discrepancy is a serious issue. Increasing the data quality should be one important goal for improving transparency and efficiency of financial markets.

With respect to data management, we plan to further investigate improvements of the HDF5 data layout, and tuning of HDF5 and filesystem parameters to improve I/O performance and efficiency of data indexing methods. While grouping the financial data by stock and date is intuitive it also leads to data fragmentation. Storing larger data portions in single arrays, while providing a virtual grouping API for fast and convenient data access could alleviate this problem.

The current version of VPIN requires all trades to be present to determine the function Φ needed for the final normalization. We plan to develop a variation of VPIN for real-time computations. It may also be fruitful to evaluate whether using volume-time for binning can improve the effectiveness of HHI.

Quantification of the effectiveness of potential early warning indicators fundamentally relies on the ability to judge whether a warning is true or false and whether relevant anomalous behavior is missed by an indicator. We, therefore, plan to develop algorithms to automatically detect anomalies in historical financial data. Evaluation of market indicators and validation of designs and implementations of real-time market monitoring systems will also benefit from realistic high-performance simulations of financial markets.

5. CONCLUSION

This “early warning” line of inquiry begins to address a key question regarding the role of high-performance computing in finance from a federal perspective: Is real-time high frequency monitoring needed? The SEC/CFTC has announced their intention to direct many billions from the financial industry to this effort, which has been criticized by others as unnecessary overkill.

We and our collaborators have come to believe that it is not overkill. Current post Flash Crash regulatory approaches are based on “circuit breakers,” which suspend trading when price or volatility triggers set them off. These are very “blunt instruments” that do not allow the market to self-correct and stabilize, and they can easily make a bad situation worse. Our tests showed that VPIN, HHI and similar indicators could provide early warning signals for a more gradual “slow down, rather than stop” replacement for on/off circuit breakers. Our HFT and academic collaborators hold this opinion strongly as well.

This work explores a number of pressing issues in implementing such an “early warning” system, such as the need for sufficient computing power to generate the warning signals and the need for reliable and effective data. We demonstrate that techniques from data-intensive sciences can address these issues. Furthermore, we believe that the same approach, likely with additional computation, are applicable in the area of financial market cyber-security, which is widely acknowledged as important, but also largely ignored in the regulatory debate.

Acknowledgments

The authors would like to thank Horst Simon for his visionary leadership in establishing the Center for Innovative Financial Technology and initiating this project; Jerry Chou, David H. Bailey, Arie Shoshani, and Orianna DeMasi for helpful discussions in planning for this work; Torrence Tran, Harrison Dekker and Masou Nikravesh for obtaining some of the data used in this project; Jeff Donovan for the Nanex data used in this work, Marcos Lopez de Prado for providing sample code and data for computing VPIN; Maureen O’Hara and David Easley for valuable discussions on VPIN; Ananth Madhavan for instructive suggestions on computing HHI in a progressive manner; Quincey Koziol for helping out organizing HDF files; Dan Weisberg for useful discussion about Panopticon. We are grateful for the insightful discussions with Gregg Berman, whose suggestions kick-started the current work.

This work is supported by the Director, Office of Laboratory Policy and Infrastructure Management of the U.S. Department of Energy under Contract No. AC02-05CH11231, and used resources of The National Energy Research Scientific Computing Center (NERSC).

6. REFERENCES

- [1] E. W. Bethel, S. Campbell, E. Dart, K. Stockinger, and K. Wu. Accelerating Network Traffic Analysis Using Query-Driven Visualization. In *Proceedings of 2006 IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122. IEEE Computer Society Press, October 2006. LBNL-59891.
- [2] CFTC/SEC. Findings regarding the market events of may 6,2010, Sep 2010. <http://www.sec.gov/news/studies/2010/marketevents-report.pdf>.
- [3] CFTC/SEC. Preliminary findings regarding the market events of may 6,2010, May 2010. www.sec.gov/sec-cftc-prelimreport.pdf.
- [4] J. Chou, K. Wu, O. Rübel, M. Howison, J. Qiang, Prabhat, B. Austin, E. W. Bethel, R. D. Ryne, and A. Shoshani. Parallel index and query for large scale data analysis. In *SC11*, 2011.
- [5] B. Curry and K. D. George. Industrial concentration: A survey. *The Journal of Industrial Economics*, 31(3):203–255, 1983.
- [6] D. Easley, N. Kiefer, M. O’Hara, and J. Paperman. Liquidity, information, and infrequently traded stocks. *Journal of Finance*, 51:1405–1436, 1996.
- [7] D. Easley, M. M. Lopez de Prado, and M. O’Hara. Flow Toxicity and Liquidity in a High Frequency World. Technical Report 9-2011, The Johnson School at Cornell University, Feb 2011.
- [8] HDF Group. HDF5 user guide, 2011. <http://hdf.ncsa.uiuc.edu/HDF5/doc/H5.user.html>.
- [9] O. C. Herfindahl. *Concentration in the US Steel Industry*. PhD thesis, Columbia University, 1950.
- [10] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft, Oct 2009.
- [11] A. O. Hirschman. *National Power and the Structure of Foreign Trade*. University of California Press, 1980.
- [12] M. Howison, Q. Koziol, D. Knaak, J. Mainzer, and J. Shalf. Tuning HDF5 for Lustre File Systems. In *Proceedings of 2010 Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS10)*, Heraklion, Crete, Greece, Sep 2010.
- [13] B. Lauterbach and U. Ben-Zion. Stock market crashes and the performance of circuit breakers: Empirical evidence. *The Journal of Finance*, 48(5):1909–1925, Dec 1993.
- [14] A. Madhavan. Exchange-traded funds, market structure and the flash crash. Technical Report in-preparation, BlackRock, 2011.
- [15] New York Stock Exchange, Inc. *TAQ 3 User’s Guide (V 1.1.9)*, Oct 2008.
- [16] M. O’Hara. *Market microstructure theory*. Blackwell, 2007.
- [17] O. Rübel, Prabhat, K. Wu, H. Childs, J. Meredith, C. G. R. Geddes, E. Cormier-Michel, S. Ahern, G. H. weber, P. Messmer, H. Hagen, B. Hamann, and E. W. Bethel. High Performance Multivariate Visual Data Exploration for Extremely Large Data. In *SuperComputing 2008 (SC08)*, Austin, Texas, USA, November 2008. LBNL-7166E.
- [18] SEC. SEC Proposes Consolidated Audit Trail System to Better Track Market Trades, SEC Press Release, May 26, 2010, <http://www.sec.gov/news/press/2010/2010-86.htm>.
- [19] K. Stockinger, E. W. Bethel, S. Campbell, E. Dart, , and K. Wu. Detecting Distributed Scans Using High-Performance Query-Driven Visualization. In *SC ’06: Proceedings of the 2006 ACM/IEEE Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, November 2006. LBNL-60053.
- [20] C. Wang, D. Chen, and M. Huang. International technology diffusion in computers & communications field. In *ICMIT*, pages 5–10, 2010.