Title

High-Performance Computational Intelligence and Forecasting Technologies

Authors

Wu, K

Simon, HD

Publication Date

2018-02-15

Peer reviewed

High-Performance Computational Intelligence and Forecasting Technologies

Kesheng Wu and Horst D. Simon

LAWRENCE BERKELEY NATIONAL LABORATORY

ONE CYCLOTRON ROAD, BERKELEY, CA 94720

High-Performance Computational Intelligence and Forecasting Technologies

Kesheng Wu and Horst D. Simon

1. Abstract

This report provides an introduction to the Computational Intelligence and Forecasting Technologies (CIFT) project at Lawrence Berkeley National Laboratory (LBNL). The main objective of CIFT is to promote the use of high-performance computing (HPC) tools and techniques for analysis of streaming data. After noticing the data volume being given as the explanation for the five-month delay for SEC and CFTC to issue their report on the 2010 Flash Crash, LBNL started the CIFT project to apply HPC technologies to manage and analyze financial data. Making timely decisions with streaming data is a requirement for many different applications, such as avoiding impending failure in the electric power grid or a liquidity crisis in financial markets. In all these cases, the HPC tools are well suited in handling the complex data dependencies and providing timely solutions. Over the years, CIFT has worked on a number of different forms of streaming data, including those from vehicle traffic, electric power grid, electricity usage, and so on. The following sections explain the key features of HPC systems, introduce a few special tools used on these systems, and provide examples of streaming data analyses using these HPC tools.

2. Introduction: Regulatory Response to the Flash Crash of 2010

On May 6, 2010, at about 2:45 p.m. (U.S. Eastern Daylight-savings Time), the U.S. stock market experienced a nearly 10% drop in the Dow Jones Industrial Average, only to recover most of the loss a few minutes later. It took about five months for regulatory agencies to come up with a formal report. In front of a congressional panel investigating the crash, the data volume (~20 terabytes) was given as the primary reason for the long delay. Since the HPC systems, such as those at National Energy Research Scientific Computing (NERSC) center,[1] routinely work with hundreds of terabytes in minutes, we should be able to process the data from financial markets. This led to the establishment of CIFT with the mission to apply the HPC techniques and tools for financial data analysis.

A key aspect of financial big data is that it consists of mostly time series. Over the years, the CIFT team, along with numerous collaborators, has developed techniques to analyze many different forms of data streams and time series. This paper provides a brief introduction to the HPC system including both hardware (Section 4) and software (Section 5), and recounts a few successful use cases (Section 6). We conclude with a summary of our vision and work so far, and also provides contact information for interested readers.

3. Background

Advances in computing technology have made it considerably easier to look for complex patterns. This pattern-finding capability is behind a number of recent scientific discoveries, such as the discovery of the Higgs particle (Aad et al. [2016]) and gravitational waves (Abbot et al. [2016]). This same capability is also at the core of many internet companies, for example, to match users with advertisers (Zeff and Aronson [1999], Yen et al. [2009]). However, the hardware and software used in science and in commerce are quite different. The HPC tools have some critical advantages that should be useful in a variety of business applications. Tools for scientists are typically built around high-performance computing (HPC) platforms, while the tools for commercial applications are built around cloud computing platforms. For the purpose of sifting through large volumes of data to find useful patterns, the two approaches have been shown to work well. However, the marquee application for HPC systems is the large-scale simulation, such as weather models used for forecasting regional storms in the next few days (Asanovic et al. [2006]), while the commercial cloud was initially motivated by the need to process a large number of independent data objects concurrently (also known as data parallel tasks).

1 NERSC is a National User Facility funded by U.S. Department of Energy, located at LBNL. More information about NERSC can be found at http://nersc.gov/.

For our work, we are primarily interested in analyses of streaming data. In particular, high-speed complex data streams, such as those from sensor network monitoring our nation's electric power grid and highway systems. This streaming workload is not well-suited for either HPC systems or cloud systems as we discuss below, but we believe that the HPC ecosystem has more to offer for streaming data analysis than the cloud ecosystem does.

Cloud systems were originally designed for data parallel tasks, where a large number of independent data objects can be processed concurrently. The system is thus designed for high throughput, not for producing real-time responses. However, many business applications require real-time or near-real-time responses. For example, an instability event in an electric power grid could develop and grow into a disaster in minutes; finding the tell-tale signature quickly enough could avert the disaster. Similarly, signs of emerging illiquidity events have been identified in the financial research literature; quickly finding these signs during the active market trading hours can offer options to prevent shocks to the market and avoid flash crashes. The ability to prioritize quick turnaround time is essential in these cases.

A data stream is by definition available progressively; therefore, there may not be a large number of data objects to be processed in parallel. Typically, only a fixed amount of the most recent data records are available for analysis. In this case, an effective way to harness the computing power of many Central Processing Units (CPUs) is to divide the analytical work on a single data object (or a single time-step) to many CPUs. The HPC ecosystem has more advanced tools for this kind of work than the cloud ecosystem does.

These are the main points that motivated our work. For a more thorough comparison of HPC systems and cloud systems, we refer interested readers to Asanovic et al. [2006]. In addition, Fox et al. [2015] have created an extensive taxonomy for describing the similarities and differences for many application scenario.

In short, we believe the HPC community has a lot to offer to advance the state-of-the-art for streaming analytics. The CIFT project was established with a mission to transfer LBNL's HPC expertise to streaming business application. We are pursuing this mission via collaboration, demonstration, and tool development. To evaluate the potential uses of HPC technology, we have spent time working with various applications. This process not only exposes our HPC experts to a variety of fields, but also makes it possible for us to gather financial support to establish a demonstration facility.

With the generous gifts from a number of early supporters of this effort, we established a substantial computing cluster dedicated to this work. This dedicated computer (named dirac1) allows users to utilize an HPC system and evaluate their applications for themselves.

We are also engaged in a tool development effort to make HPC systems more usable for streaming data analysis. In the following sections, we will describe the hardware and software of the dedicated CIFT machine, as well as some of the demonstration and tool development efforts. Highlights include improving the data handling speed by 21-fold, and increasing the speed of computing an early warning indicator by 720-fold.

4. HPC Hardware

Legend has it that the first generation of big data systems was built with the spare computer components gleaned from a university campus. This is likely a urban legend, but it underscores an important point about the difference between HPC systems and cloud systems. Theoretically, a HPC system is built with custom high-cost components, while cloud systems are built with standard low-cost commodity components. In reality, the worldwide investment in HPC systems is much smaller than that of personal computers, there is little incentive for computer manufacturers to produce custom components just for the HPC market. The HPC systems are largely assembled from commodity components just like cloud systems. However, due to their different target applications, there are some differences in their choices of the components.

Here, we describe the computing elements, storage system, and networking system in turn. Figure 1 is a high-level schematic diagram representing the key components of the Magellan cluster around year 2010 (Jackson et al. [2010]; Yelick et al. [2011]). The computer elements include both CPUs and Graphics Processing Units (GPUs). These CPUs and GPUs are commercial products in all cases, for example, the nodes on dirac1 use a 24-core 2.2Ghz Intel processor, which is common to cloud computing systems. Currently, dirac1 does not use GPUs.

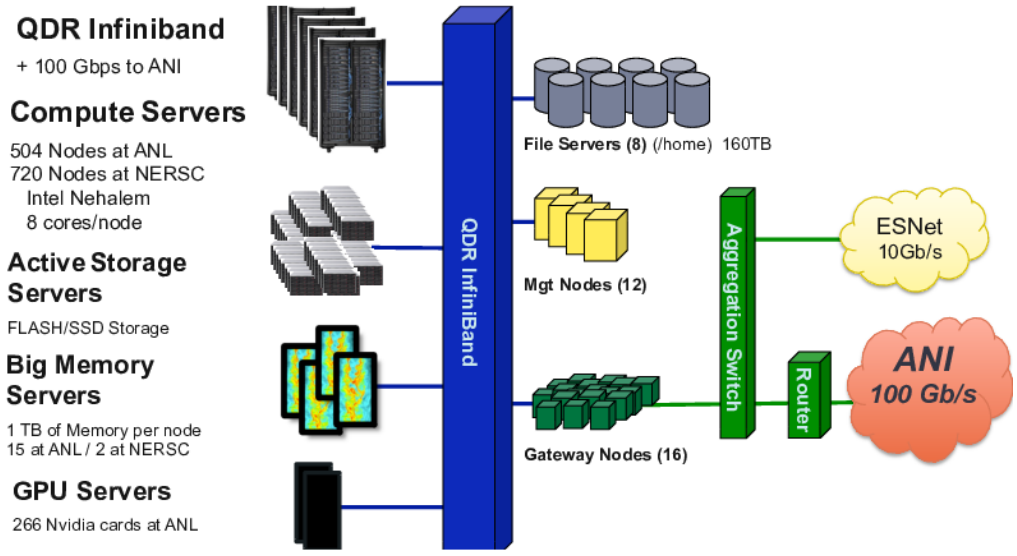

Figure 1: Schematic of the Magellan cluster (circa 2010), an example of HPC computer cluster.

In Figure 1, the networking system consists of two parts: the InfiniBand network connecting the components within the cluster, and the switched network connection to the outside world. In this particular example, the outside connections are labeled "ESNet" and "ANI". The InfiniBand network switches are also common in cloud computing systems.

The storage system in Figure 1 includes both rotating disks and flash storage. This combination is also common. What is different is that a HPC system typically has its storage system concentrated outside of the computer nodes, while a typical cloud computing system has its storage system distributed among the compute nodes. These two approaches have their own advantages and disadvantages. For example, the concentrated storage is typically exported as a global file system to all compute nodes, which makes it easier to deal with data stored in files. However, this requires a highly capable network connecting the CPUs and the disks. In contrast, the distributed approach could use lower-capacity network because there is some storage that is close to each CPU. Typically, a distributed file system, such as the Google file system (Ghemawat, Gobioff, and Leung [2003]), is layered on top of a cloud computing system to make the storage accessible to all CPUs.

In short, the current generation of HPC systems and cloud systems use pretty much the same commercial hardware components. Their differences are primarily in the arrangement of the storage systems and networking systems. Clearly, the difference in the storage system designs could affect the application performance. However, the virtualization layer of the cloud systems is likely the bigger cause of application performance difference. In the next section, we will discuss another factor that could have an even larger impact, namely software tools and libraries.

Virtualization is generally used in the cloud computing environment to make the same hardware available to multiple users and to insulate one software environment from another. This is one of the more prominent features distinguishing the cloud computing environment from the HPC environment. In most cases, all three basic components of a computer system—CPU, storage, and networking—are all virtualized. This virtualization has many benefits. For example, an existing application can run on a CPU without recompiling; many users can share the same hardware; hardware faults could be corrected through the virtualization software; and applications on a failed compute node could be more easily migrated to another node. However, this virtualization layer also imposes some runtime overhead that would reduce application performance. For time-sensitive applications, this reduction in performance could become a critical issue.

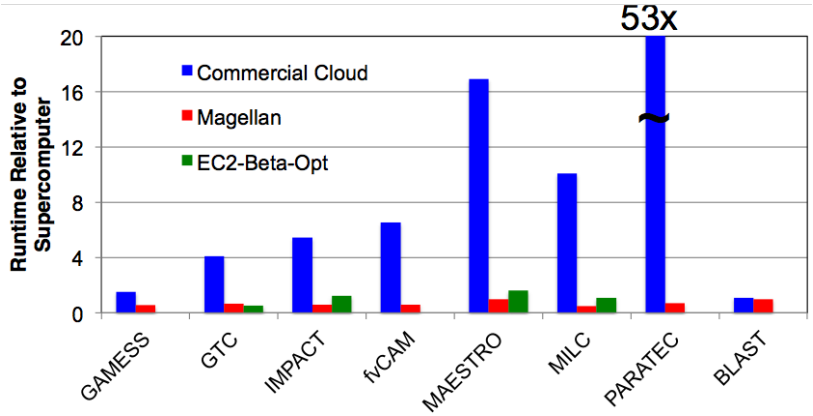

Figure 2: Cloud ran scientific applications considerably slower than on HPC systems (circa 2010)

Tests show that the performance differences could be quite large. Next, we briefly describe a performance study reported by Jackson et al [2010]. Figure 2 shows the performance slowdown using different computer systems. The names below the horizontal axis are different software packages commonly used at NERSC. The left bar corresponds to the Commercial Cloud, the middle bar corresponds to Magellan, and the (sometimes missing) right bar corresponds to the EC2-Beta-Opt system. The non-optimized commercial cloud instances run these software packages 2 to 10 times slower than on Magellan. Even on the more expensive high-performance instances, there are noticeable slowdowns.

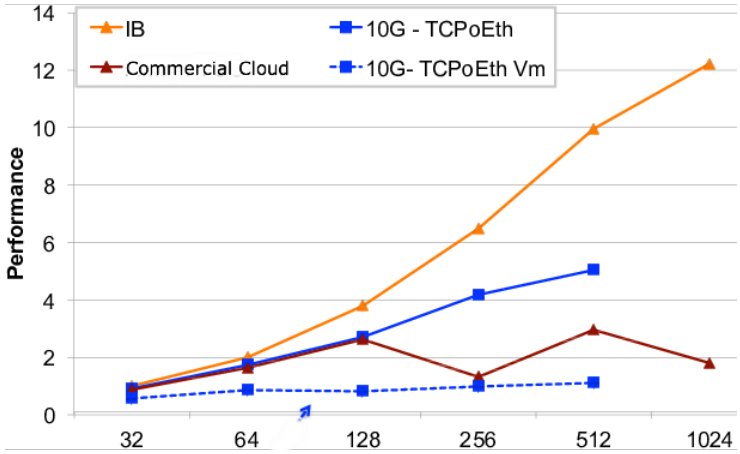

Figure 3: As the number of cores increases (horizontal axis), the virtualization overhead becomes much more significant (circa 2010)

Figure 3 shows a study of the main factor causing the slowdown with the software package PARATEC. In Figure 2, we see that PARATEC took 53 times longer to complete on the commercial cloud than on an HPC system. We observe from Figure 3 that, as the number of cores (horizontal axis) increases, the differences among the measured performances (measured in TFLOP/s) become larger. In particular, the line labeled "10G- TCPoEth Vm" barely increases as the number of cores grows. This is the case where the network instance is using virtualized networking (TCP over Ethernet). It clearly shows that the networking virtualization overhead is significant, to the point of rendering the cloud useless.

The issue of virtualization overhead is widely recognized (Chen et al. [2015]). There has been considerable research aimed at addressing both the I/O virtualization overhead (Gordon et al. [2012]) as well as the networking virtualization overhead (Dong et al. [2012]). As these state-of-the-art techniques are gradually moved into commercial products, we anticipate the overhead will decrease in the future, but some overhead will inevitably remain.

To wrap up this section, we briefly touch on the economics of HPC versus cloud. Typically, HPC systems are run by nonprofit research organizations and universities, while cloud systems are provided by commercial

companies. Profit, customer retention, and many other factors affect the cost of a cloud system (Armburst et al. [2010]). In 2011, the Magellan project report stated that "Cost analysis shows that DOE centers are cost competitive, typically 3–7x less expensive, when compared to commercial cloud providers" (Yelick et al. [2010]).

A group of high-energy physicists thought their use case was well-suited for cloud computing and conducted a detailed study of a comparison study (Holzman et al. [2017]). Their cost comparisons still show the commercial cloud offerings as approximately 50% more expensive than dedicated HPC systems for comparable computing tasks; however, had the authors worked with a larger (and more realistic) data set, the cost of data ingress and egress would incur hefty charges and make the cost using the cloud much higher. For complex workloads, such as the streaming data analyses discussed in this report, we anticipate that this HPC cost advantage will remain in the future. A 2016 National Academy of Sciences study came to the same conclusion that even a long-term lease from Amazon is likely 2 to 3 times more expensive than HPC systems to handle the expected science workload from NSF (Box 6.2 from National Academies of Sciences, [2016]).

5. HPC Software

Ironically, the real power of a supercomputer is in its specialized software. There are a wide variety of software packages available for both HPC systems and cloud systems. In most cases, the same software package is available on both platforms. Therefore, we chose to focus on software packages that are unique to HPC systems and have the potential to improve computational intelligence and forecasting technologies. One noticeable feature of the HPC software ecosystem is that much of the application software performs its own interprocessor communication through Message Passing Interface (MPI). In fact, the cornerstone of most scientific computing tools is MPI (Kumar et al. [1994], Gropp, Lusk, and Skjellum [1999]). Accordingly, our discussion of HPC software tools will start with MPI. Following that, we will briefly review data management tools (Shoshami and Rotem [2010]).

5.1. Message Passing Interface

Message Passing Interface (MPI) is a communication protocol for parallel computing (Gropp, Lusk, and Skjellum [1999], Snir et al. [1988]). It defines a number of point-to-point data exchange operations as well as some collective communication operations. The MPI standard was established based on several early attempts to build portable communication libraries. An early implementation from Argonne National Lab, named MPICH, was highly efficient, scalable, and portable. It helped MPI to gain wide acceptance among scientific users.

The success of MPI is partly due to its separation of Language Independent Specifications (LIS) from its language bindings. This allows the same core function to be provided to many different programming languages, which also contributes to its acceptance. The first MPI standard specified ANSI C and Fortran-77 bindings together with the LIS. The draft specification was presented to the user community at the 1994 Supercomputing Conference.

Another key factor contributing to MPI's success is the open-source license used by MPICH. This license allows the vendors to take the source code to produce their own custom versions, which allows the HPC system vendors to quickly produce their own MPI libraries. To this day, all HPC systems support the familiar MPI on their computers. This wide adoption also ensures that MPI will continue to be the favorite communication protocol among the users of HPC systems.

5.2. Hierarchical Data Format

In describing the HPC hardware components, we noted that the storage systems in an HPC platform are typically different from those in a cloud platform. The software libraries used by most users for accessing the storage systems are also different due to the difference in the conceptual models of data. Typically, HPC applications treat data as multi-dimensional arrays and, therefore, the most popular I/O libraries on HPC systems are designed to work with multi-dimensional arrays. Here, we describe the most widely used array format library, HDF5 (Folk et al. [2011]).

HDF5 is the fifth iteration of the Hierarchical Data Format, produced by the HDF Group.[2] The basic unit of data in HDF5 is an array plus its associated information such as attributes, dimensions, and data type. Together, they are known as a data set. Data sets can be grouped into a larger unit called a group, and groups can be organized into high-level groups. This flexible hierarchical organization allows users to express complex relationships among the data sets.

Beyond the basic library for organizing user data into files, the HDF Group also provides a suite of tools and specialization of HDF5 for different applications. For example, HDF5 includes a performance profiling tool. NASA has a specialization of HDF5, named HDF5-EOS, for their data from Earth-Observing System (EOS); and the next-generation DNA sequence community has produced a specialization named BioHDF for their bioinformatics data.

HDF5 provides an efficient way for accessing the storage systems on HPC platform. In tests, we have demonstrated that using HDF5 to store stock markets data significantly speeds up the analysis operations. This is largely due to its efficient compression/decompression algorithms that minimize network traffic and I/O operations, which brings us to our next point.

5.3. *In Situ* Processing

Over the last few decades, CPU performance has roughly doubled every 18 months (Moore's law), while disk performance has been increasing less than 5% a year. This difference has caused it to take longer and longer to write out the content of the CPU memory. To address this issue, a number of research efforts have focused on *in situ* analysis capability (Ayachit et al. [2016]).

Among the current generation of processing systems, the Adaptable I/O System (ADIOS) is the most widely used (Liu et al. [2014]). It employs a number of data transport engines that allow users to tap into the I/O stream and perform analysis operations. This is useful because irrelevant data can be discarded in-flight, hence avoiding its slow and voluminous storage. This same *in situ* mechanism also allows it to complete write operations very quickly. In fact, it initially gained attention because of its write speed. Since then, the ADIOS developers have worked with a number of very large teams to improve their I/O pipelines and their analysis capability.

Because ADIOS supports streaming data accesses, it is also highly relevant to CIFT work. In a number of demonstrations, ADIOS with ICEE transport engine was able to complete distributed streaming data analysis in real-time (Choi et al. [2013]). We will describe one of the use cases involving blobs in fusion plasma in the next section.

In short, *in situ* data processing capability is another very useful tool from the HPC ecosystem.

5.4. Convergence

We mentioned earlier that the HPC hardware market is a tiny part of the overall computer hardware market. The HPC software market is even smaller compared to the overall software market. In many cases, the HPC software tools are largely maintained by a number of small vendors along with some open-source contributors. Therefore, HPC system users are under tremendous pressure to migrate to the better supported cloud software systems. This is a significant driver for convergence between software for HPC and software for cloud (Fox et al. [2015]).

Even though convergence appears to be inevitable, we advocate for a convergence option that keeps the advantage of the software tools mentioned above. One of the motivations of the CIFT project is to seek a way to transfer the above tools to the future computing environment.

6. Use Cases

Data processing is such an important part of modern scientific research that some researchers are calling it the fourth paradigm of science (Hey, Tansley, and Tolle [2009]). In economics, the same data-driven research activities have led to the wildly popular behavioral economics (Camerer and Loewenstein [2011]). Much of the recent advances in data-driven research are based on machine learning applications (Qiu et al. [2016], Rudin and Wagstaff [2014]). Their successes in a wide variety of fields, such as planetary science and

2 The HDF Group web site is https://www.hdfgroup.org/.

bioinformatics, have generated considerable interest among researchers from diverse domains. In the rest of this section, we describe a few examples applying advanced data analysis techniques to various fields, where many of these use cases originated in the CIFT project.

6.1. Supernova Hunting

In astronomy, the determination of many important facts such as the expansion speed of the universe, is performed by measuring the light from exploding type Ia supernovae (Bloom et al. [2012]). The process of searching the night sky for exploding supernovae is called synoptic imaging survey. The Palomar Transient Factory (PTF) is an example of such a synoptic survey (Nicholas et al. [2009]). The PTF telescopes scan the night sky and produce a set of images every 45 minutes. The new image is compared against the previous observations of the same patch of sky to determine what has changed and to classify the changes. Such identification and classification tasks used to be performed by astronomers manually. However, the current number of incoming images from the PTF telescopes is too large for manual inspection. An automated workflow for these image processing tasks has been developed and deployed at a number of different computer centers.

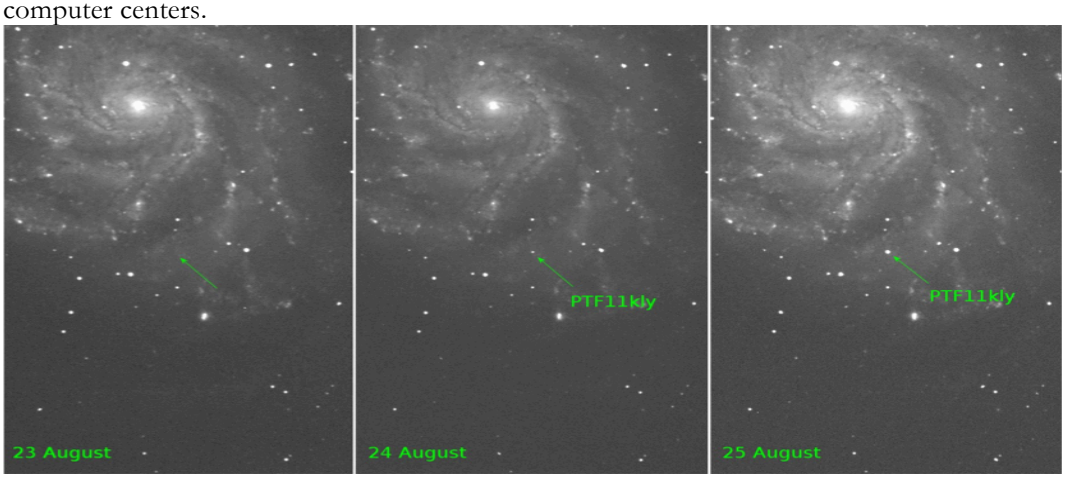

Figure 4: Supernova SN 2011fe was discovered 11 hours after first evidence of explosion, as a result of the extensive automation in classification of astronomical observations

Figure 4 shows the supernova that was identified earliest in its explosion process. On August 23, 2011, the patch of the sky showed no sign of this star, but a faint light showed up on August 24. This quick turnover allowed astronomers around the world to perform detailed follow-up observations, which are important for determining the parameters related to the expansion of the universe.

The quick identification of this supernova is an important demonstration of the machine learning capability of the automated workflow. This workflow processes the incoming images to extract the objects that have changed since last observed. It then classifies the changed object to determine a preliminary type based on the previous training. Since follow-up resources for extracting novel science from fast-changing transients are precious, the classification not only needs to indicate the assumed type, but also the likelihood and confidence of the classification. Using classification algorithms trained on PTF data, the mislabeling of transients and variable stars has a 3.8% overall error rate, at >96% classification accuracy. Additional work is expected to achieve higher accuracy rates in upcoming surveys, such as the Large Synoptic Survey Telescope.

6.2. Blobs in Fusion Plasma

Large-scale scientific exploration in domains such as physics and climatology are huge international collaborations involving thousands of scientists each. As these collaborations produce more and more data at progressively faster rates, the existing workflow management systems are hard-pressed to keep pace. A necessary solution is to process, analyze, summarize, and reduce the data before it reaches the relatively slow disk storage system, a process known as in-transit processing (or in-flight analysis). Working with the ADIOS developers, we have implemented the ICEE transport engine to dramatically increase the data-handling capability of collaborative workflow systems (Choi et al. [2013]). This new feature significantly improved the data flow management for distributed workflows. Tests showed that the ICEE engine allowed a number of

large international collaborations to make near real-time collaborative decisions. Here, we briefly describe the fusion collaboration involving KSTAR.

KSTAR is a nuclear fusion reactor with fully superconducting magnets. It is located in South Korea, but there are a number of associated research teams around the world. During a run of a fusion experiment, some researchers are controlling the device at KSTAR, but others want to participate by performing collaborative analysis of the preceding runs of the experiment to provide advice on how to configure the device for the next run. During the analysis of the experimental measurement data, scientists might run simulations or examine previous simulations to study parameter choices. Typically, there may be a lapse of 10 to 30 minutes between two successive runs, and all collaborative analyses need to complete during this time window in order to affect the next run.

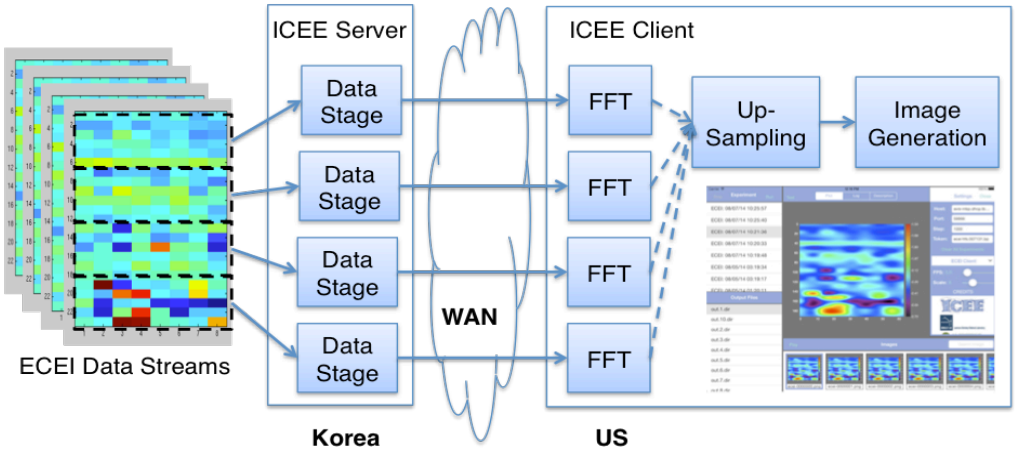

Figure 5: A distributed workflow for studying fusion plasma dynamics

We have demonstrated the functionality of the ICEE workflow system with two different types of data: one from the Electron Cyclotron Emission Imaging (ECEI) data obtained at KSTAR, and the other involving synthetic diagnostic data from the XGC modeling. The distributed workflow engine needs to collect data from these two sources, extract a feature known as blobs, track the movement of these blobs, predict the movement of the blobs in the experimental measurements, and then provide advices on actions to be performed. Figure 5 shows how the ECEI data is processed. The workflow for the XGC simulation data is similar to what is shown in Figure 5, except that the XGC data is located at NERSC.

To be able to complete the above analytical tasks in real-time, effective data management with ICEE transport engine of ADIOS is only part of the story. The second part is to detect blobs efficiently (Wu et al. [2016]). In this work, we need to reduce the amount of data transported across wide-area networks by selecting only the necessary chunks. We then identify all cells within the blobs and group these cells into connected regions in space, where each connected region forms a blob. The new algorithm we developed partitions the work onto different CPU cores by taking full advantage of the MPI for communication between the nodes and the shared memory among the CPU cores on the same node. Additionally, we also updated the connected component label algorithm to correctly identify blobs at the edge, which were frequently missed by the earlier detection algorithms. Overall, our algorithm was able to identify blobs in a few milliseconds for each time step by taking full advantage of the parallelism available in the HPC system.

6.3. Intraday Peak Electricity Usage

Utility companies are deploying advanced metering infrastructure (AMI) to capture electricity consumption in unprecedented spatial and temporal detail. This vast and fast-growing stream of data provides an important testing ground for the predictive capability based on big data analytical platforms (Kim et al. [2015]). These cutting-edge data science techniques, together with behavioral theories, enable behavior analytics to gain novel insights into patterns of electricity consumption and their underlying drivers (Todd et al. [2014]).

As electricity cannot be easily stored, its generation must match consumption. When the demand exceeds the generation capacity, a blackout will occur, typically during the time when consumers need electricity the most.

Because increasing generation capacity is expensive and requires years of time, regulators and utility companies have devised a number of pricing schemes intended to discourage unnecessary consumption during peak demand periods.

To measure the effectiveness of a pricing policy on peak demand, one can analyze electricity usage data generated by AMI. Our work focuses on extracting baseline models of household electricity usage for a behavior analytics study. The baseline models would ideally capture the pattern of household electricity usage including all features except the new pricing schemes. There are numerous challenges in establishing such a model. For example, there are many features that could affect the usage of electricity but for which no information is recorded, such as the temperature set point of an air-conditioner or the purchase of a new appliance. Other features, such as outdoor temperature, are known, but their impact is difficult to capture in simple functions.

We developed a number of new baseline models that could satisfy the above requirements. At present, the gold standard baseline is a well-designed randomized control group. We showed that our new data-driven baselines could accurately predict the average electricity usage of the control group. For this evaluation, we use a well-designed study from a region of the United States where the electricity usage is the highest in the afternoon and evening during the months of May through August. Though this work concentrates on demonstrating that the new baseline models are effective for groups, we believe that these new models are also useful for studying individual households in the future.

We explored a number of standard black-box approaches. Among machine learning methods, we found gradient tree boosting (GTB) to be more effective than others. However, the most accurate GTB models require lagged variables as features (for example, the electricity usage a day before and a week before). In our work, we need to use the data from year T-1 to establish the baseline usage for year T and year T+1. The lagged variable for a day before and a week before would be incorporating recent information not in year T-1. We attempted to modify the prediction procedure to use the recent predictions in place of the actual measured values a day before and a week before; however, our tests show that the prediction errors accumulate over time, leading to unrealistic predictions a month or so into the summer season. This type of accumulation of prediction errors is common to continuous prediction procedures for time series.

To address the above issue, we devised a number of white-box approaches, the most effective of which, known as LTAP, is reported here. LTAP is based on the fact that the aggregate variable electricity usage per day is accurately described by a piece-wise linear function of average daily temperature. This fact allows us to make predictions about the total daily electricity usage. By further assuming the usage profile of each household remains the same during the study, we are able to assign the hourly usage values from the daily aggregate usage. This approach is shown to be self-consistent; that is, the prediction procedure exactly reproduces the electricity usage in year T–1, and the predictions for the control group in both year T and T+1 are very close to the actual measured values. Both treatment groups have reduced electricity usages during the peak-demand hours and the active group reduced the usage more than the passive group. This observation is in line with other studies.

Though the new data-driven baseline model LTAP predicts the average usages of the control group accurately, there are some differences in predicted impact of the new time-of-use pricing intended to reduce the usage during the peak-demand hours (see Figure 6). For example, with the control group as the baseline, the active group reduces its usage by 0.277 kWh (out of about 2 kWh) averaged over the peak-demand hours in the first year with the new price and 0.198 kWh in the second year. Using LTAP as the baseline, the average reductions are only 0.164 kWh for both years. Part of the difference may be due to the self-selection bias in treatment groups, especially the active group, where the households have to explicitly opt-in to participate in the trial. It is likely that the households that elected to join the active group are well-suited to take advantage of the proposed new pricing structure. We believe that the LTAP baseline is a way to address the self-selection bias, and plan to conduct additional studies to further verify this.

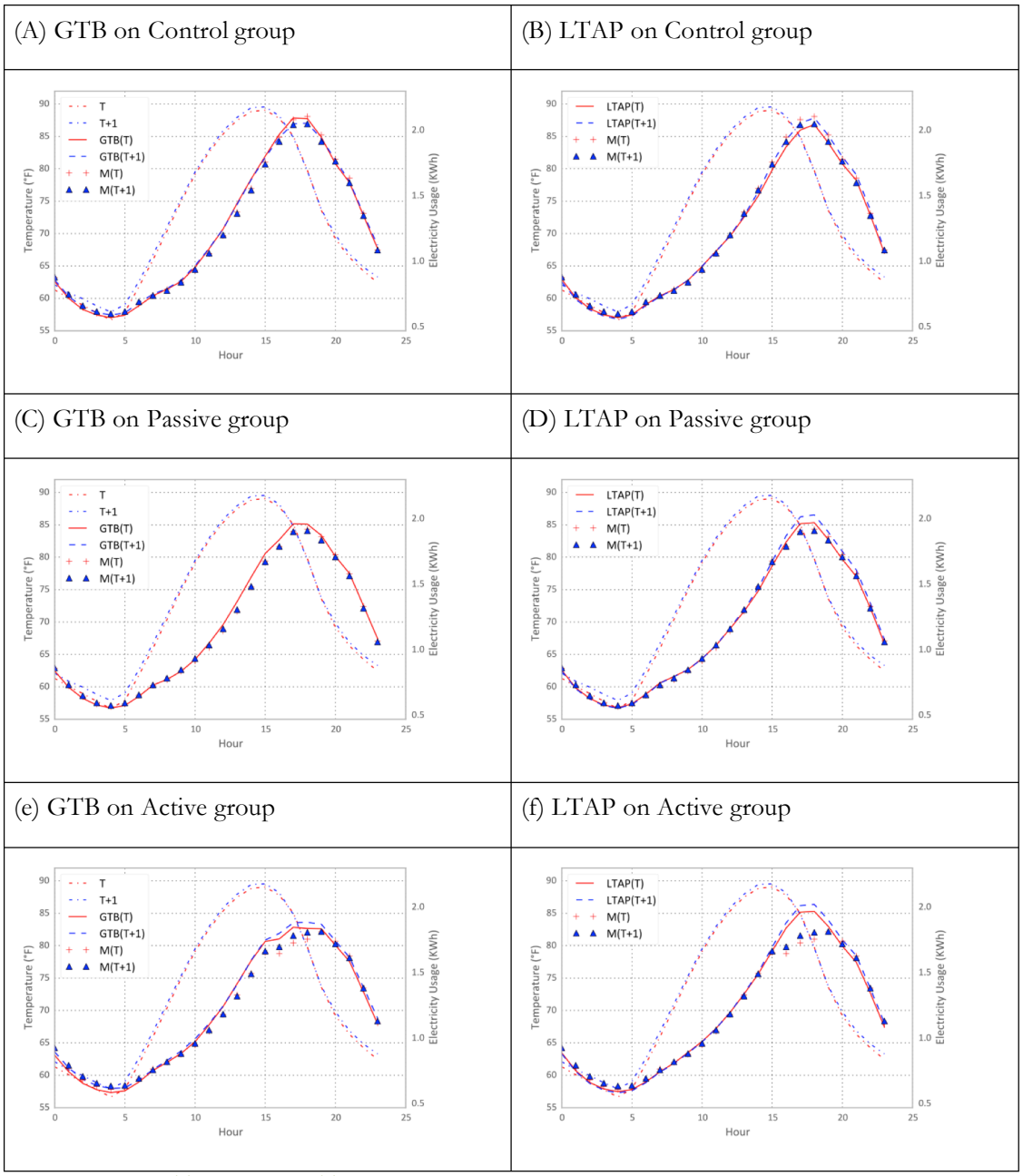

Figure 6: Gradient Tree Boost (GBT) appears to follow recent usage too closely and therefore not able to predict the baseline usage as well as the newly develop method named LTAP.

6.4. The Flash Crash of 2010

The extended time it took for the SEC and CFTC to investigate the Flash Crash of 2010 was the original motivation for CIFT's work. Federal investigators needed to sift through tens of terabytes of data to look for the root cause of the crash. Since CFTC publicly blamed the volume of data to be the source of the long delay, we started our work by looking for HPC tools that could easily handle tens of terabytes. Since HDF5 is the most commonly used I/O library, we started our work by applying HDF5 to organize a large set of stock trading data (Bethel at al. [2011]).

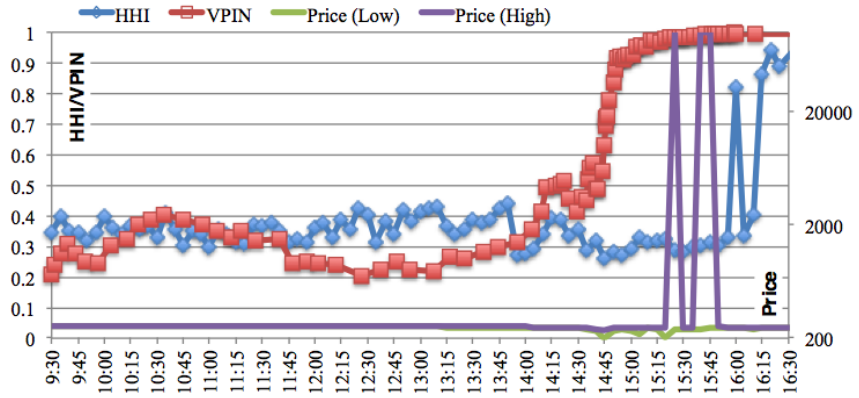

Figure 7: Apple Stock price on May 6, 2010, along with HHI and VPIN values computed every 5 minutes during the market hours

Let us quickly review what happened during the 2010 Flash Crash. On May 6, at about 2:45 p.m. (U.S. Eastern Daylight Time), the Dow Jones Industrial Average dropped almost 10%, and many stocks were traded at one cent per share, the minimum price for any possible trade. Figure 7 shows an example of another extreme case, where shares of Apple (symbol AAPL) traded at $100,000 per share, the maximum possible price allowed by the exchange. Clearly, these were unusual events, which undermined investors' faith and confidence in our financial markets. Investors demanded an answer to what caused these events. To make our work relevant to the financial industry, we sought to experiment with the HDF5 software, and apply it to the concrete task of computing earlier warning indicators. Based on recommendations from a group of institutional investors, regulators, and academics, we implemented two sets of indicators that have been shown to have "early warning" properties preceding the Flash Crash. They are the Volume Synchronized Probability of Informed Trading (VPIN) (Easley, Lopez de Prado, and O'Hara [2011]) and a variant of the Herfindahl-Hirschman Index (HHI) (Hirschman [1980]) of market fragmentation. We implemented these two algorithms in the C++ language, while using MPI for inter-processor communication, to take full advantage of the HPC systems. The reasoning behind this choice is that if any of these earlier warning indicators is shown to be successful, the high-performance implementation would allow us to extract the warning signals as early as possible so there might be time to take corrective actions. Our effort was one of the first steps to demonstrate that it is possible to compute the earlier warning signals fast enough.

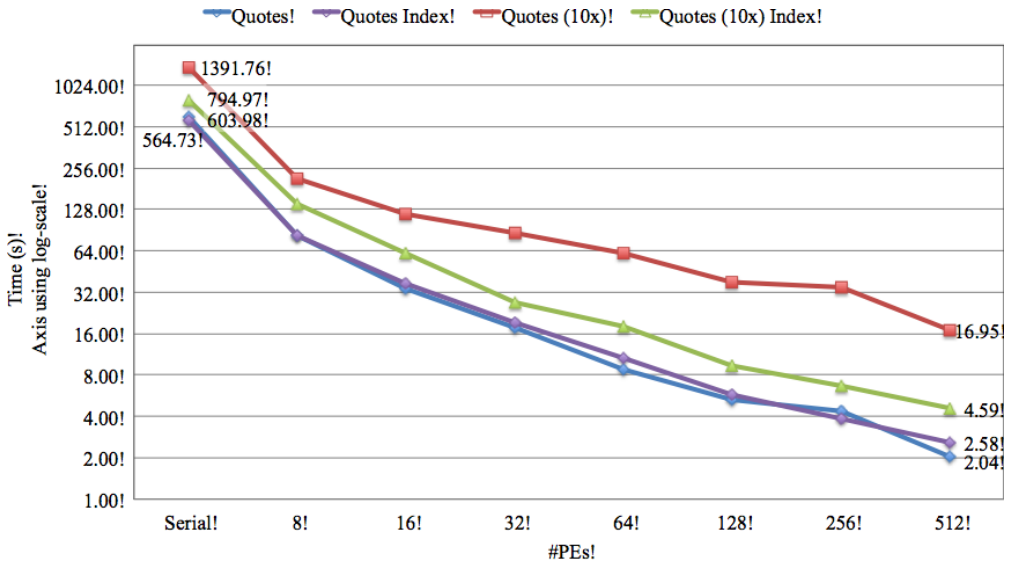

Figure 8: Time to process 10-year worth of SP500 quotes data stored in HDF5 files, which takes 21 times longer when the same data is in ASCII files (603.98 seconds versus approximately 3.5 hours)

For our work, we implemented two versions of the programs: one uses data organized in HDF5 files, and another reads the data from the commonly used ASCII text files. Figure 8 shows the time required to process the trading records of all S&P 500 stocks over a 10-year timespan. Since the size of the 10-year trading data is still relatively small, we replicated the data ten times as well. On a single CPU core (labeled" "Serial" in Figure 8), it took about 3.5 hours with ASCII data, but only 603.98 seconds with HDF5 files. When 512 CPU cores are used, this time reduces to 2.58 seconds using HDF5 files, resulting in a speedup of 234 times.

On the larger (replicated) dataset, the advantage of HPC code for computing these indices is even more pronounced. With ten times as much data, it took only about 2.3 times longer for the computer to complete the tasks, a below-linear latency increase. Using more CPU makes HPC even more scalable.

Figure 8 also shows that with a large data set, we can further take advantage of the indexing techniques available in HDF5 to reduce the data access time (which in turn reduces the overall computation time). When 512 CPU cores are used, the total runtime is reduced from 16.95 seconds to 4.59 seconds, a speedup of 3.7 due to this HPC technique of indexing.

6.5. Volume-synchronized Probability of Informed Trading Calibration

Understanding the volatility of the financial market requires the processing of a vast amount of data. We apply techniques from data-intensive scientific applications for this task, and demonstrate their effectiveness by computing an early warning indicator called Volume Synchronized Probability of Informed Trading (VPIN) on a massive set of futures contracts. The test data contains 67 months of trades for the hundred most frequently traded futures contracts. On average, processing one contract over 67 months takes around 1.5 seconds. Before we had this HPC implementation, it took about 18 minutes to complete the same task. Our HPC implementation achieves a speedup of 720 times.

Note that the above speedup was obtained solely based on the algorithmic improvement, without the benefit of parallelization. The HPC code can run on parallel machines using MPI, and thus is able to further reduce the computation time.

The software techniques employed in our work include the faster I/O access through HDF5 described above, as well as a more streamlined data structure for storing the bars and buckets used for the computation of VPIN. More detailed information is available in Wu et al. [2013].

With a faster program to compute VPIN, we were also able to explore the parametric choices more closely. For example, we were able to identify the parameter values that reduce VPIN's false positive rate over one hundred contracts from 20% to only 7%, see Figure 9. The parameter choices to achieve this performance are: (1) pricing the volume bar with the median prices of the trades (not the closing price typically used in analyses), (2) 200 buckets per day, (3) 30 bars per bucket, (4) support window for computing VPIN = 1 day,

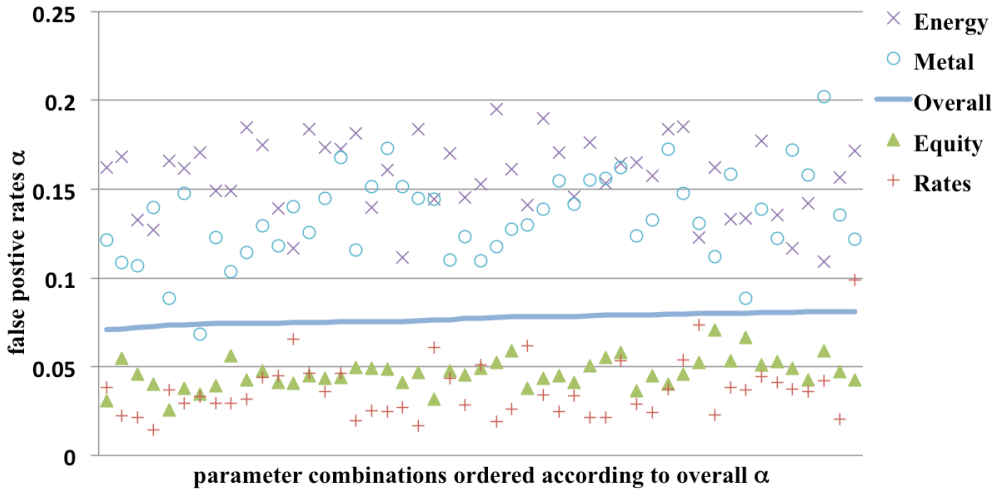

Figure 9: The average false positive rates (α) of different classes of futures contracts ordered according to their average.

event duration = 0.1 day, (5) bulk volume classification with Student t-distribution with $\nu = 0.1$, and (6) threshold for CDF of VPIN = 0.99. Again, these parameters provide a low false positive rate on the totality of futures contracts, and are not the result of individual fitting.

On different classes of futures contracts, it is possible to choose different parameters to achieve even lower false positive rates. In some cases, the false positive rates can fall significantly below 1%. Based on Figure 9, interest rate and index futures contracts typically have lower false positive rates. The futures contracts on commodities, such as energy and metal, generally have higher false positive rates.

Additionally, a faster program for computing VPIN allows us to validate that the events identified by VPIN are "intrinsic," in the sense that varying parameters such as the threshold on VPIN CDF only slightly change the number of events detected. Had the events been random, changing this threshold from 0.9 to 0.99 would have reduced the number of events by a factor of 10. In short, a faster VPIN program also allows us to confirm the real-time effectiveness of VPIN.

6.6. Revealing High Frequency Events with Non-uniform Fast Fourier Transform

High Frequency Trading is pervasive across all electronic financial markets. As algorithms replace tasks previously performed by humans, cascading effects similar to the 2010 Flash Crash may become more likely. In our work (Song et al. [2014]), we brought together a number of high performance signal-processing tools to improve our understanding of these trading activities. As an illustration, we summarize the Fourier analysis of the trading prices of natural gas futures.

Normally, Fourier analysis is applied on uniformly spaced data. Since market activity comes in bursts, we may want to sample financial time series according to trading activity. For example, VPIN samples financial series as a function of volume traded. However, a Fourier analysis of financial series in chronological time may still be instructive. To this purpose, we use a non-uniform FFT procedure.

From the Fourier analysis of the natural gas futures market, we see strong evidences of High Frequency Trading in the market. The Fourier components corresponding to high frequencies are (1) becoming more prominent in the recent years and (2) are much stronger than could be expected from the structure of the market. Additionally, a significant amount of trading activity occurs in the first second of every minute, which is a tell-tale sign of trading triggered by algorithms that target a Time-Weighted Average Price (TWAP).

Fourier analysis on trading data shows that activities at the once-per-minute frequency are considerably higher than neighboring frequencies (see Figure 10). Note that the vertical axis is in logarithmic scale. The strength of activities at once-per-minute frequency is more than ten times stronger than the neighboring frequencies. Additionally, the activity is very precisely defined at once-per-minute, which indicates that these trades are triggered by intentionally constructed automated events. We take this to be strong evidence that TWAP algorithms have a significant presence in this market.

We expected the frequency analysis to show strong daily cycles. In Figure 10, we expect amplitude for frequency 365 to be large. However, we see the highest amplitude was for the frequency of 366. This can be explained because 2012 was a leap year. This is a validation that the non-uniform FFT is capturing the expected signals. The second- and third-highest amplitudes have the frequencies of 732 and 52, which correspond to activities happening twice-a-day and once-a-week. These are also unsurprising.

We additionally applied the non-uniform FFT on the trading volumes and found further evidence of algorithmic trading. Moreover, the signals pointed to a stronger presence of algorithmic trading in recent years. Clearly, non-uniform FFT algorithm is useful for analyzing highly irregular time series.

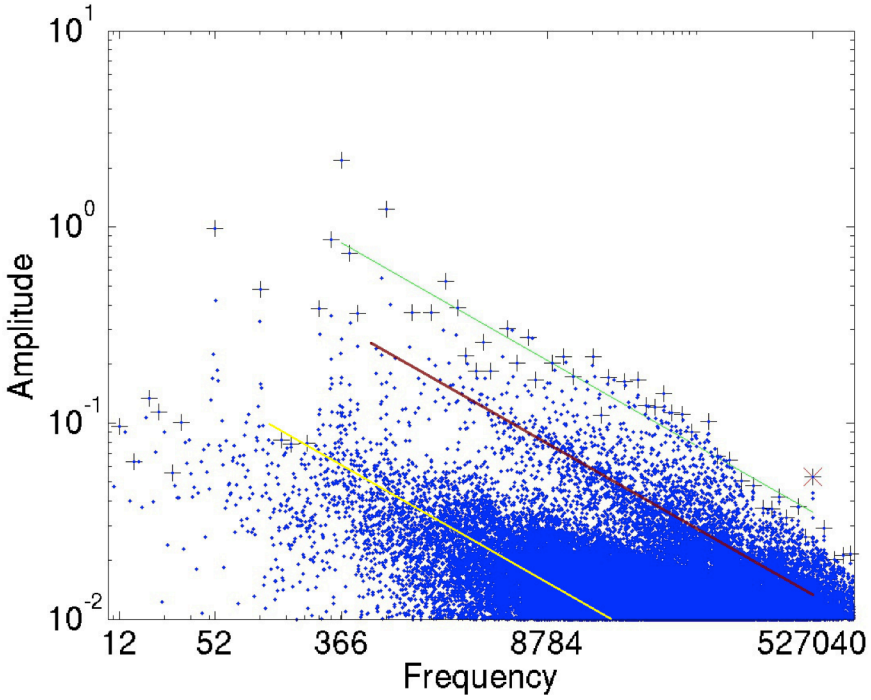

Figure 10: Fourier spectrum of trading prices of natural gas futures contracts in 2012. Non-uniform FFT identifies strong presence of activities happening once per day (frequency = 366), twice per day (frequency = 732), and once per minute (frequency = 527040 = 366*24*60).

7. Summary and Call for Participation

Currently, there are two primary ways to construct large-scale computing platforms: the HPC approach and the cloud approach. Most of the scientific computing efforts use the HPC approach; while most of the business computing follows the cloud approach. The conventional wisdom is that the HPC approach occupies a small niche of little consequence. This is not true. HPC systems are essential to the progress of scientific research. They played important roles in exciting new scientific discoveries including the Higgs particle and gravitational waves. They have spurred the development of new subjects of study, such as behavioral economics, and new ways of conducting commerce through the Internet. The usefulness of extremely large HPC systems has led to the 2015 National Strategic Computing Initiative.[3]

There are efforts to make HPC tools even more useful by accelerating their adoption in business applications. The HPC4Manufacturing[4] effort is pioneering this knowledge transfer to the U.S. manufacturing industry, and has attracted considerable attention. Now is the time to make a more concerted push for HPC to meet other critical business needs.

We have developed CIFT to show that a broad class of business applications could benefit from the HPC tools and techniques. In decisions such as how to respond to a voltage fluctuation in a power transformer and an early warning signal of impending market volatility event, HPC software tools could help determine the signals early enough for decision makers, provide sufficient confidence about the prediction, and anticipate the consequence before the catastrophic event arrives. These applications have complex

3 The National Strategic Computing Initiative plan is available online at https://www.whitehouse.gov/sites/whitehouse.gov/files/images/NSCI%20Strategic%20Plan.pdf. The Wikipedia page on this topic https://en.wikipedia.org/wiki/National_Strategic_Computing_Initiative also has some useful links to additional information.

4 Information about HPC4Manufacturing is available online at https://hpc4mfg.llnl.gov/.

computational requirements and often have a stringent demand on response time as well. HPC tools are better suited to meet these requirements than cloud-based tools.

In our work, we have demonstrated that the HPC I/O library HDF5 can be used to accelerate the data access speed by 21-fold, and HPC techniques can accelerate the computation of the Flash Crash early-warning indicator VPIN by 720-fold. We have developed additional algorithms that enable us to predict the daily peak electricity usage years into the future. We anticipate that applying HPC tools and techniques to other applications could achieve similarly significant results.

In addition to the performance advantages mentioned above, a number of published studies (Yelick et al. [2011], Holzman et al. [2017]) show HPC systems to have a significant price advantage as well. Depending on the workload's requirement on CPU, storage, and networking, using a cloud system might cost 50% more than using a HPC system, and, in some cases, as much as seven times more. For the complex analytical tasks described in this report, with their constant need to ingest data for analysis, we anticipate the cost advantage will continue to be large.

CIFT is expanding the effort to transfer HPC technology to private companies, so that they can also benefit from the price and performance advantages enjoyed by large-scale research facilities. Our earlier collaborators have provided the funds to start a dedicated HPC system for this work. This resource should make it considerably easier for interested parties to try out their applications on an HPC system. We are open to different forms of collaborations. For further information regarding CIFT, please visit CIFT's web page at `http://crd.lbl.gov/cift/` or contact the authors.

8. Acknowledgments

The CIFT project is the brainchild of Dr. David Leinweber. Dr. Horst Simon brought it to LBNL in 2010. Drs. E. W. Bethel and D. Bailey led the project for four years.

The CIFT project has received generous gifts from a number of donors. This work is supported in part by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research also uses resources of the National Energy Research Scientific Computing Center supported under the same contract.

9. References

1. Aad, G., et al. (2016): "Measurements of the Higgs boson production and decay rates and coupling strengths using pp collision data at $$\sqrt{s}=7$$ s = 7 and 8 TeV in the ATLAS experiment." *The European Physical Journal C*, Vol. 76, No. 1, p. 6.
2. Abbott, B.P. et al. (2016): "Observation of gravitational waves from a binary black hole merger." *Physical Review Letters,* Vol. 116, No. 6, p. 061102.
3. Zeff, R.L. and B. Aronson (199): *Advertising on the Internet.* New York: John Wiley & Sons, Inc.
4. Yan, J. et al. (2009): "How much can behavioral targeting help online advertising?" Proceedings of the 18th international conference on world wide web. ACM: Madrid, Spain. pp. 261–270.
5. Asanovic, K. et al. (2006): "The landscape of parallel computing research: A view from Berkeley." *Technical Report UCB/EECS-2006-183*, EECS Department, University of California, Berkeley.
6. Fox, G. et al. (2015): "Big Data, simulations and HPC convergence, iBig Data benchmarking": 6th International Workshop, WBDB 2015, Toronto, ON, Canada, June 16–17, 2015; and 7th International Workshop, WBDB 2015, New Delhi, India, December 14–15, 2015, Revised Selected Papers, T. Rabl, et al., editors. 2016, Springer International Publishing: Cham. pp. 3–17.DOI:10.1007/978-3-319-49748-8_1.
7. Jackson, K. R. et al. (2010): "Performance analysis of high performance computing applications on the Amazon Web Services Cloud. *Cloud Computing Technology and Science (CloudCom),* 2010 IEEE Second International Conference on. 2010, IEEE.
8. Yelick, K., et al. (2011): "The Magellan report on cloud computing for science." U.S. Department of Energy, Office of Science.
9. Ghemawat, S., H. Gobioff, and S.-T. Leung (2003): "The Google file system," *SOSP '03: Proceedings of the nineteenth ACM symposium on operating systems principles.* ACM: Bolton Landing, NY. pp. 29–43.

10. National Academies of Sciences, Engineering and Medicine (2016): *Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017–2020*. Washington, DC: The National Academies Press.
11. Chen, L. et al. (2015): "Profiling and understanding virtualization overhead in cloud." *Parallel Processing (ICPP)*, 2015 44th International Conference on, IEEE.
12. Gordon, A. et al. (2012): "ELI: Bare-metal performance for I/O virtualization." S*IGARCH Comput. Archit. News*, Vol. 40, No. 1, pp. 411–422.
13. Dong, Y. et al. (2012): "High performance network virtualization with SR-IOV." *Journal of Parallel and Distributed Computing*, Vol. 72, No. 11, pp. 1471–1480.
14. Armbrust, M., et al. (2010): "A view of cloud computing." *Communications of the ACM*, Vol. 53, No. 4, pp. 50–58.
15. Holzman, B. et al. (2017): "HEPCloud, a new paradigm for HEP facilities: CMS Amazon Web Services investigation. *Computing and Software for Big Science,* Vol. 1, No. 1, p. 1.
16. Kumar, V. et al. (1994): *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Redwood City, California: The Benjamin/Cummings Publishing Company.
17. Gropp, W., E. Lusk, and A. Skjellum (1999): *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA: MIT Press.
18. Shoshani, A. and D. Rotem (2010): "Scientific Data Management : Challenges, Technology, and Deployment." *Chapman & Hall/CRC Computational Science Series*. 2010, Boca Raton: CRC Press.
19. Snir, M. et al. (1998): *MPI: The Complete Reference. Volume 1, The MPI-1 Core*. Cambridge, MA: MIT Press.
20. Folk, M. et al. (2011): "An overview of the HDF5 technology suite and its applications." Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases. ACM.
21. Ayachit, U. et al. "Performance analysis, design considerations, and applications of extreme-scale in situ infrastructures." Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE Press.
22. Liu, Q. et al., (2014): "Hello ADIOS: The challenges and lessons of developing leadership class I/O frameworks." *Concurrency and Computation: Practice and Experience,* Volume 26, No. 7, pp. 1453–1473.
23. Choi, J.Y. et al. (2013): ICEE: "Wide-area in transit data processing framework for near real-time scientific applications." 4th SC Workshop on Petascale (Big) Data Analytics: Challenges and Opportunities in Conjunction with SC13.
24. Hey, T., S. Tansley, and K.M. Tolle (2009): *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Vol. 1. Microsoft research Redmond, WA.
25. Camerer, C.F. and G. Loewenstein (2011): "Behavioral economics: Past, present, future." In *Advances in Behavioral Economics,* p. 1--52.
26. Qiu, J. et al. (2016): "A survey of machine learning for big data processing." *EURASIP Journal on Advances in Signal Processing,* Vol. 2016, No. 1, p. 67. DOI: 10.1186/s13634-016-0355-x
27. Rudin, C. and K. L. Wagstaff (2014) "Machine learning for science and society." *Machine Learning,* Vol. 95, No. 1, pp. 1–9.
28. Bloom, J. S. et al. (2012): "Automating discovery and classification of transients and variable stars in thesynoptic survey era." *Publications of the Astronomical Society of the Pacific*, Vol. 124, No. 921, p. 1175.
29. Nicholas, M. L. et al. (2009): "The Palomar transient factory: System overview, performance, and first results." *Publications of the Astronomical Society of the Pacific,* Vol. 121, No. 886, p. 1395.
30. Wu, L. et al. (2016): "Towards real-time detection and tracking of spatio-temporal features: Blob-filaments in fusion plasma. I*EEE Transactions on Big Data,* Vol. 2, No. 3, pp. 262–275.
31. Kim, T. et al. (2015): "Extracting baseline electricity usage using gradient tree boosting." IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE.
32. Todd, A. et al. (2014): *Insights from Smart Meters: The Potential for Peak Hour Savings from Behavior-Based Programs*. Lawrence Berkeley National Laboratory. https://www4.eere.energy.gov/seeaction/system/files/documents/smart_meters.pdf
33. Bethel, E. W. et al. (2011): "Federal market information technology in the post Flash Crash era: Roles for supercomputing." Proceedings of WHPCF'2011, ACM: New York, NY. pp. 23–30.

34. Easley, D., M. Lopez de Prado, and M. O'Hara (2011): "The microstructure of the 'Flash Crash': Flow toxicity, liquidity crashes and the probability of informed trading. *Journal of Portfolio Management,* Vol. 37, No. 2, pp. 118–128.
35. Hirschman, A. O. (1980): National Power and the Structure of Foreign Trade. Vol. 105. University of California Press.
36. Wu, K. et al. (2013): *A Big Data Approach to Analyzing Market Volatility. Algorithmic Finance. v 2, no. 3, pp241--267. 2013.*
37. Song, J. H. et al. (2014): "Exploring irregular time series through non-uniform fast Fourier transform." Proceedings of the 7th Workshop on High Performance Computational Finance, IEEE Press.