White Paper
August 2014

# IT@Intel
# Maximizing Marketing Insight through Big Data Analytics

Intel IT and the Corporate Marketing Group believe that the marketing analytics platform and its underlying technology will deliver significant business value to the enterprise.


**Richard Mason**
Marketing Analytics Product Owner, Intel IT

**Seshu Edala**
Capability Engineer, Intel IT

**Navneet Kumar**
BI Developer, Big Data, Intel IT

**S. Suman**
BI Developer, Big Data, Intel IT

**Pradeep Baluvaneralu**
BI Analyst, Intel IT

**Adapa N. K. Eswara Reddy**
BI Developer, Big Data, Intel IT

**Lakshmanan Letchumanan**
Info Tech Manager, Intel IT


## Executive Overview


An important use case for the Intel® Data Platform, built on Apache Hadoop* software, is providing insight and enabling optimization of Intel's marketing investments. The goal is to enable near-real-time insights into marketing campaign performance and to point to optimization opportunities. In the past, Intel has typically analyzed marketing data from one media channel at a time. Our solution creates an opportunity to analyze several media channels simultaneously to produce greater insight into how multiple marketing channels are working in combination in the integrated campaign.

Intel IT and the Corporate Marketing Group believe that the marketing analytics platform and its underlying technology will deliver significant business value to the enterprise through maximizing effectiveness and efficiency of future marketing campaigns.

Intel IT chose the Intel Data Platform as the enabling technology for the marketing analytics platform. A key component was to investigate what technical enhancements to the Intel Data Platform were necessary to support the marketing analytics platform. The technical objectives included the following:

- Reduce data processing time using available new technologies
- Expose the data to make it available for quicker discovery

Using several optimizing techniques available within the Intel Data Platform, we have achieved benefits relating to processing time and storage space. Our Apache Spark* and Apache Shark* proofs of concept have provided additional optimization in these areas.

The project team believes that the marketing analytics platform and its underlying technology will deliver significant business value to the enterprise. Our strategy will provide better focus to future marketing campaigns, enabling Intel to invest in the campaign approaches that deliver the greatest engagement, brand awareness, and demand generation.

Contents



Contributor

**Tatiana Shusterman**
Manager, CMG Insights Marketing Research

Acronyms

**ETL** extract, transform, load
**MPP** massively parallel processing
**RDD** resilient distributed dataset

# Background

To address Intel's need for big data analytics, Intel IT implemented the Intel® Data Platform, based on Apache Hadoop* software. Multiple business groups use this platform to uncover the hidden value in big data. One such group is the Corporate Marketing Group, which has started to use the Intel Data Platform to analyze marketing campaign performance using multiple data sources, including data from both paid and owned media channels (paid search, paid digital banners, paid social investments, Intel.com, and owned social media). Historically, the analysis has examined one media channel at a time, with little integration between channels.

Integrating the analysis of multiple media channels can provide better and faster insights into a customer journey and achieve the ultimate goal to drive greater engagement with Intel's audiences, improving brand awareness and driving demand generation.

Figure 1 illustrates an opportunity to expand data analysis across multiple channels, understanding the impact that the different paid portions of a campaign have on driving customer engagement and optimally balancing the investment across the channels. An additional analysis will be to look at the impact that one channel has on another, for example, consumers exposed through one channel (paid digital banner) and then engaging through a second (paid/organic search).

To technically support such a solution, Intel IT needed to evaluate its implementation of the Intel Data Platform (see A Look at Today's Intel® Data Platform sidebar). In the platform's current version, data processing of a single media channel consumes 4 to 5 hours for daily processing and 12 hours for 18 months of historical processing. Given the increase in variety and volume of data, and the complexity of integrating this data across channels, the platform requires optimization to keep the daily processing at a manageable level. The platform also requires further support for interactive querying, which will enable the marketing group to ask more complex questions.

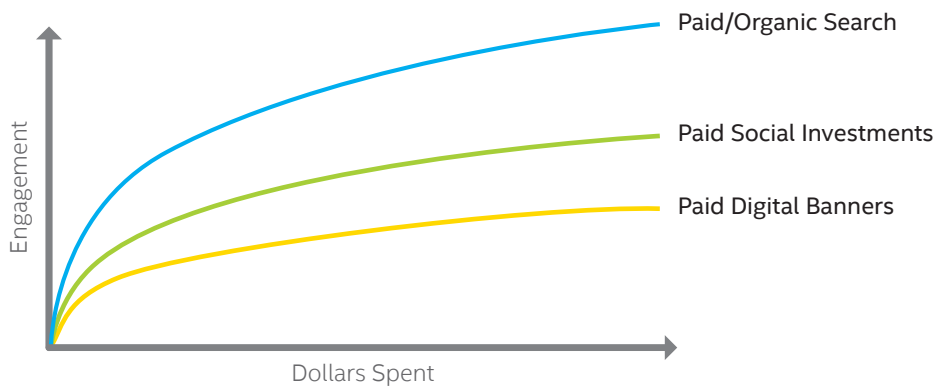



Figure 1. The marketing group wants to expand campaign analysis to include data across all media channels, such as paid digital banners, paid social investments, and paid/organic search. This approach will give us a more comprehensive understanding of campaign performance and help us define where investments are working best and how to optimize those investments.

# Solution

Intel IT and the Corporate Marketing Group formed a project team and embarked on creating a marketing analytics platform. The project team selected the Intel Data Platform as the basis for the technical solution because of its ability to collect, align, analyze, and discover insights using disparate data sources. The marketing analytics platform that the project team is building has the following business objectives:

- Enable business executives worldwide to have timely access to data
- Create a detailed, integrated view of the customer journey including how customers travel from one channel to another on their paths to purchase
- Provide a flexible platform for advanced analytics, data mining, and historical trends

A key component to achieving the business objectives is to implement the right technical environment to support these objectives. This process involved enabling enhancements to the Intel Data Platform and investigating how best to apply emerging technologies with the goal of achieving the following:

- Reduce data processing time using available new technologies
- Expose the data to make it available for quicker and more flexible discovery

## Marketing Analytics Platform

The project team envisions a single data and reporting platform that provides rich analytics capabilities to the marketing group. As described in Table 1, this platform will enable the following:

- **Campaign performance.** Standardize performance measurements for marketing campaigns and content across channels (paid, owned, and earned) and automate reporting across media channels.
- **Campaign optimization.** Drive internal, near-real-time analytics to optimize marketing campaigns, including cross-channel campaigns.
- **Analytics sandbox.** Provide a sandbox for data mining and advanced analytics opportunities.

We believe that the platform will deliver the following business benefits:

- Allocate marketing spend more effectively within and across media channels
- Improve investment allocation decisions through further optimization of marketing content, channels, and campaigns

In the future, the solution should help align the marketing and sales organizations by creating a view of how exposure and engagement with Intel marketing translates into sales.

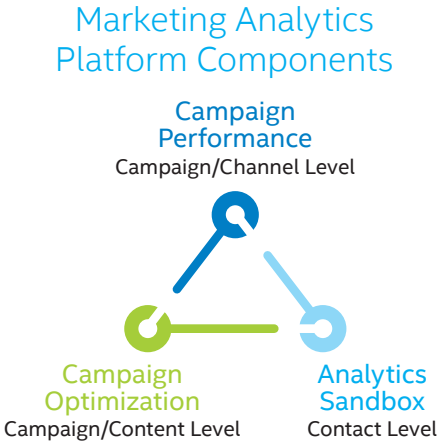



Table 1. The marketing analytics platform will enable the marketing group to assess campaign effectiveness within and across channels.

| | |
|---|---|
| Campaign Performance | **Campaign/Channel Level**<br>• How many exposure opportunities did a particular campaign generate?<br>• What was the overall engagement rate?<br>• What channel generated higher engagement per dollar spent? |
| Campaign Optimization | **Campaign/Content Level**<br>• What content created higher engagement per dollar spent in a specific channel for a campaign?<br>• If a piece of content was used across multiple channels, where did it perform better?<br>• What paid components of a media campaign had the highest impact on owned and earned media? |
| Analytics Sandbox | **Contact Level**<br>• What are the demographics and behavioral profiles of customers who shop on Intel.com or respond to Intel's ads? Where and how can the marketing group retarget them?<br>• What is the correlation between various types of engagement and shopping on Intel.com? |

## Intel® Data Platform Architecture

The project team was in a unique position within Intel, holding dual responsibilities for both defining the technology (the enabling platform and its distribution) and defining the business value (customer requirements) of the marketing analytics program. Instead of simply matching requirements to a platform, the team used storyboarding to identify all near-term use cases, possibilities, and limitations of the Intel Data Platform technology. Then the team provided these requirements to Intel IT. Intel IT was tasked with broadening the technology stack beyond what was immediate, because we were also planning for the future.

To support the marketing analytics platform, Intel IT identified that the solution needs to address the three types of data processing:

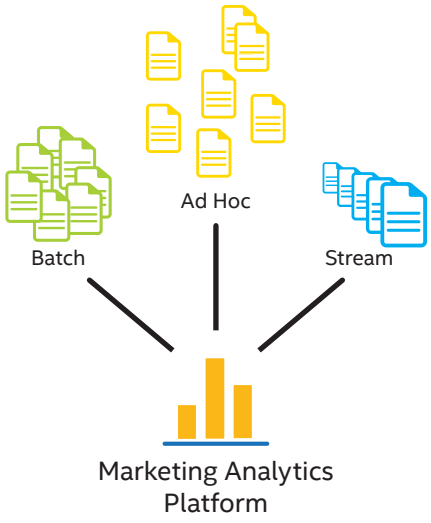
- **Batch processing.** A long-running analytics cycle that processes massive amounts of data at rest, primarily transforming raw data into reportable data.
- **Interactive "ad hoc" querying.** Queries that enable users to discover descriptive and prescriptive insights by combining factual, dimensional, and raw data.
- **Stream processing.** A data processing pipeline that continuously processes and analyzes small quantities of data as it arrives before storing it on disk.

Each of these types of data processing serves different big data characteristics, including volume, velocity, value, and variability. The current version of the Intel Data Platform can effectively perform batch processing and interactive querying; the batch processing applies massively parallel transformations to voluminous web log data at rest while the Hive Open Database Connectivity database interface supports the ad hoc querying of the datasets by business users. However, the ad hoc query using Hive can take several seconds, so it does not provide a true interactive experience.

Our vision for the marketing analytics platform requires the addition of stream processing as well as interactive analysis to our solution to achieve near-real-time insights into the data, reducing both the data latency and the query latency from when data transpires to when actionable insights form. To do so, we evaluated our current data flow architecture to identify where we needed optimizations to support stream processing and interactive querying.

In the future, the solution should help align the marketing and sales organizations by creating a view of how exposure and engagement with Intel marketing translates into sales.

Data Processing Types




**Data Flow Architecture**

As shown in Figure 2, the current Intel Data Platform implementation follows this data flow:

1. Data flows in from various sources, such as social feeds and web data.
2. Apache Camel* routes the data into the Intel Data Platform using enterprise integration patterns.
3. Apache MapReduce*, Apache Pig*, and Apache Hive* process the data using transformation rules:
   - **MapReduce.** A distributed computation feature that coordinates each of the servers in the cluster to operate on part of the overall processing task in parallel.
   - **Pig.** An interactive scripting environment used for data cleansing, filtering, and transformations.
   - **Hive.** An SQL language used for dimensional joins and aggregations. It is also used for ad hoc querying.
4. Online analytical processing cubes are connected directly to Hadoop using the Hive Open Database Connectivity interface to reduce data hops.
5. Using native utilities and connectors, Apache Sqoop* imports and exports data from Hadoop to a relational massively parallel processing (MPP) database. The relational MPP database enables interactive reporting and querying.

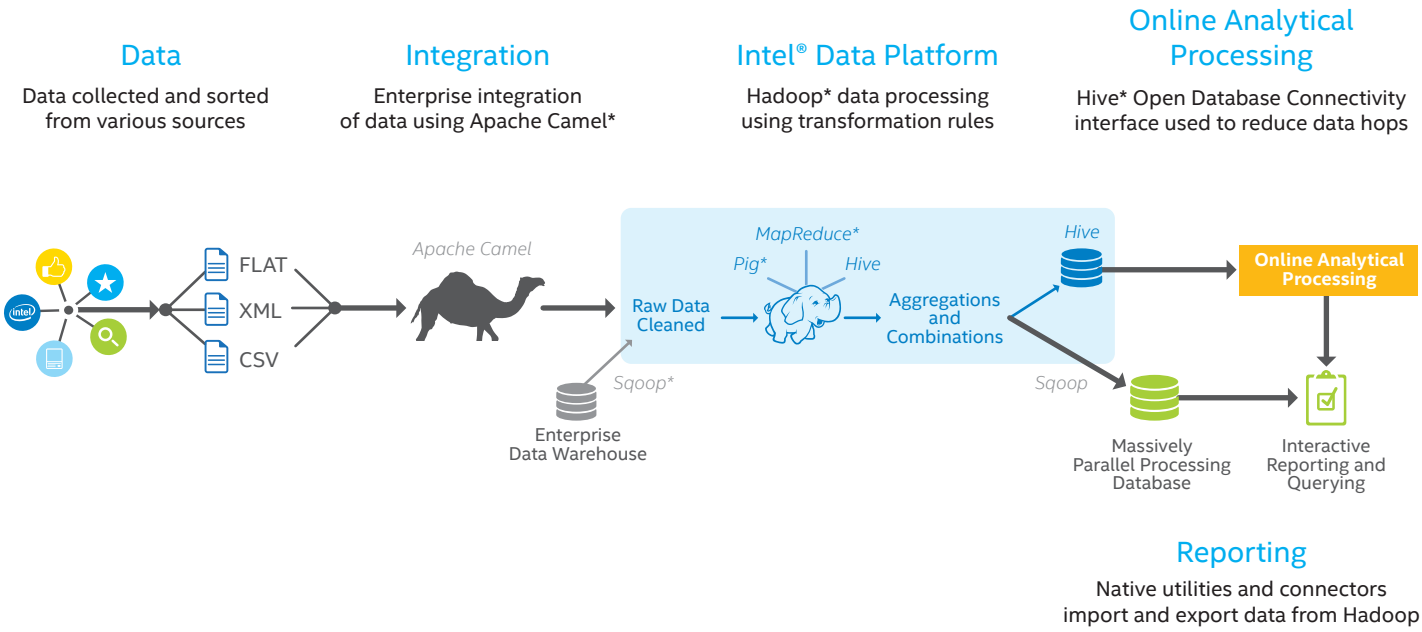



Figure 2. Data from various sources flows into the Intel® Data Platform. The Intel Data Platform then transforms the data for analysis and reporting. Within the Intel Data Platform, we are taking steps to reduce processing time through the use of partitioning schemes and other optimizations.

**Optimizing the Intel Data Platform**

Using optimizing techniques available within the Intel Data Platform, we have achieved benefits in processing time and storage space.

For large files in Hadoop, we used the SequenceFile format and enabled compression to minimize I/O latency and avoid numerous short mapper tasks. For aggregated data in Hadoop to be exported to the MPP database, we used an uncompressed text format for compatibility, resulting in storage sizes that were four to five times smaller. This optimization reduced our end-to-end execution duration in production by 2.5 hours.

For all intermediate files in terabytes, we considered increasing the block size. Smaller block sizes can mean redundant setup and teardown costs for short tasks. Following best practices in the industry for handling terabytes, we chose to increase the block size to 256 MB.

We also decided to monitor all jobs on a daily basis, so that we can adjust the number of reducers as the data volume increases over time. Instead of relying solely on Hive and Pig, we are incorporating subject matter expertise and data knowledge to determine the optimal number of reducers to eliminate unnecessary overhead.

We also chose to use the RCFile columnar format to store Hive tables. At any point in time, only a few columns are operated on for business rules. Columnar storage, combined with partitioning, allows for optimal data layout, access, storage, and compression of data. This yields reduced storage requirements and maximal processing throughput.

# Proofs of Concept

In addition to the initial optimizations mentioned previously, we investigated Apache Spark* and Apache Shark* with the goal of further enhancing our Intel Data Platform implementation and enabling it to handle stream processing.

## Apache Spark Proof of Concept

Apache Spark is an MPP system built with the Scala* programming language. Spark accomplishes distributed computing by abstracting data into fault-tolerant, resilient distributed datasets (RDDs). RDDs are computed and cached in memory on demand, and are partition-, locality-, and lineage-aware. RDDs are also immutable.

The key difference between MapReduce and RDDs is the transformation sequence: MapReduce is disk-based and transforms data using sequential map-reduce processes, whereas RDDs compute data on demand from the lineage and keep that data persistent in memory until it is deemed invalid or unnecessary. If a memory fault occurs and makes the data invalid, RDDs reapply the full data lineage to rematerialize the data in memory.

## A Look at Today's Intel® Data Platform

The Intel® Data Platform is an open source software product designed to enable a wide range of data analytics on Apache Hadoop* software. The platform is optimized for Apache Hive* queries, provides connectors for the open source R* statistical programming language, and enables graph analytics using Intel® Graph Builder for Apache Hadoop software—a library to convert large datasets into graphics to help visualize relationships between data.

Other key features include the following:

- A boost in Hadoop performance through optimizations for Intel® Xeon® processors and Intel® 10GbE Server Adapters
- Data confidentiality through encryption and decryption performed without a performance penalty in the storage layer—the Hadoop Distributed File System*—taking full advantage of enhancements provided by Intel® Advanced Encryption Standard New Instructions[1]
- Role-based access control with cell-level granularity available through HBase*, an open source, nonrelational distributed database that runs on top of the Hadoop Distributed File System
- Multisite scalability and adaptive data replication enabled through HBase and the Hadoop Distributed File System
- Up to a 3.5x improvement in Hive query performance

[1] Intel® AES-NI requires a computer system with an AES-NI enabled processor, as well as non-Intel software to execute the instructions in the correct sequence. AES-NI is available on select Intel® processors. For availability, consult your reseller or system manufacturer. For more information, see Intel® Advanced Encryption Standard Instructions (AES-NI).

Spark, by design, is faster than traditional MapReduce for several reasons, including the following:

- The data is priority persisted in memory instead of written to disk.
- Frequently accessed data can optionally be cached in memory and reused across multiple transformation chains.
- Because RDDs are immutable and strongly typed, record sets can be indexed, which provides superior performance for both indexed access and block access to records.

We conducted a proof of concept to determine if the batch processing routines in Pig can be refactored in Spark to gain performance benefits. We tested various sizes of datasets, ranging from 65 KB to 400 MB. We investigated the effects of data serialization on memory management and query performance, and whether in-memory compression was beneficial. Our Spark proof of concept led us to identify several best-known methods, described in Table 2.

## Apache Shark Proof of Concept

Apache Shark is an in-memory SQL query engine that runs on top of Spark and uses Hive. Shark is intended to replace the Hive layer with two potential advantages:

- Shark can bypass traditional MapReduce and leverage Spark's RDD models for faster data processing.
- The Shark server can act as the central cache that multiple client sessions can leverage.

In parallel with our Spark proof of concept, we conducted another proof of concept to evaluate Shark's reduced latency claims. Our objective was to determine if Shark's RDD models and the in-memory benefits were substantial enough to deliver an interactive concurrent user experience.

We designed two queries:

- An ad hoc query that a reporting environment would typically issue when connecting to a Hive system. This query employs a simple star schema to join a fact table with multiple dimensional tables. This query used a 75 MB fact table.
- A simple data warehousing ETL (extract, transform, and load) query that transforms a raw dataset into a reportable one. The query operated over a 10 GB dataset.

Table 2. Spark* Best-Known Methods

| Best-Known Method | Key Driver |
|---|---|
| Kryo serialization | Records from RDDs need to be serialized and deserialized on the network. We found that using Kryo serialization provides more than twice the space savings compared to Java* serialization. Kryo imposes a small overhead over the native Java object representation but is significantly faster than Java serialization. |
| Resilient distributed dataset (RDD) compression | If data does not fit in memory using Kryo serialization, consider using RDD compression. Note that the size in memory will be reduced at the expense of speed. There is a small degradation in performance. |
| Snappy* compression | Consider using Snappy compression instead of the default LZF compression. While our tests indicated faster performance for Snappy, we did not specifically measure the compression savings. |
| Garbage collection | Consider storing data in Kryo serialized format in memory as a single large byte array to help reduce garbage collection pauses. The higher the number of Java objects, the higher the cost of garbage collection. Because the number of objects stored is fewer after serialization, the cost of garbage collection is significantly reduced. |
| Primitives | Because Java objects consume significant space in memory, consider using primitives instead where applicable. |
| Columnar storage of data on disk | To store data on disk, consider using columnar storage such as RCfile or Parquet. Parquet provides better compression and columnar format representation than RCfile, enabling faster access to subsets of columns. Additionally, the compression reduces I/O. |
| Co-partitioning | Consider using co-partitioning to help ensure that the aggregations and joins happen on the local node without having to do a shuffle or network transfer. This approach enables jobs to run faster. |
| Disk cache for RDDs | Consider using the local disk for caching RDDs. Use sparingly if the depth of the RDDs is high; that is, if the number of transformations in the lineage from the source to the destination is small and local, the cost of reconstruction may be smaller than deserializing from disk. |

For both queries, we found that the results were similar: Shark delivered results that were five to six times faster than Hive (see Figure 3). However, we found that the in-memory benefits were not as substantial. We formed the hypothesis that the in-memory benefits did not prove out in either case for the following reasons:

- The ad hoc query does not allow for any substantial reuse of the cached immutable datasets. Also, the dataset is too small to merit a substantial difference reading from memory compared to disk.
- The ETL scenario involved a significant amount of data, which can slow down in-memory processing. Also, the data is transformed during the query without much lasting benefit, since subsequent queries do not take advantage of that transformation.

Overall, we believe that Shark is a worthwhile enhancement over the current Hive layer. However, it does not provide an interactive discovery experience. For a more interactive experience, we plan to next evaluate Cloudera* Impala and PrestoDB*, which are both SQL query engines that run in Hadoop.

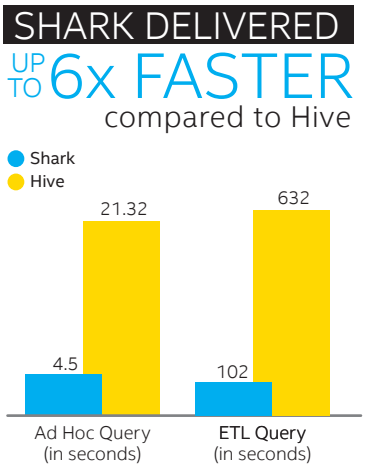



Figure 3. Shark* executed our ad hoc and ETL (extract, transform, load) queries five to six times faster than Hive*.

# Next Steps

In addition to optimizing the current big data batch processing with Spark and Shark and reducing the query latency using Impala or PrestoDB, our marketing analytics platform needs near-real-time analytics that will enable us to gain insights as data arrives. Our current daily data ingestion schedules are too latent for real-time analytics.

We plan to continue to investigate the use of the following components to determine whether they can decrease the latency from when data is ready to when it converts into a decision point:

- **Apache Flume*.** A fault-tolerant distributed log aggregation system for handling large incremental log files.
- **Apache Kafka*.** A fast, scalable, distributed, high-throughput messaging service for numerous loosely connected publishers and subscribers, enabling them to exchange data in sync.
- **Apache Storm*.** A distributed concurrent data processing pipeline that simplifies complex ETL topologies on continuous, perpetual, unbounded streams of data.

Together the three components combine to form a continuous data collection, communication, and processing system that enables rapid-but-small data to complement slow-but-big data systems.

# Conclusion

The Corporate Marketing Group and Intel IT are deploying a marketing analytics program that will help Intel to allocate marketing spend more effectively within and across media channels. Central to this program is a marketing analytics platform that will enable the marketing group to analyze data across channels, providing a broader picture of how content, channels, and campaigns are working in combination. Intel IT's proofs of concept have yielded information that is helping the marketing group align the technical solution with the program goals.

The project team is continuing to develop the marketing analytics platform and implement enhancements to the Intel Data Platform, anticipating that the solution will provide significant benefits to the marketing and sales organizations. Intel IT also expects to apply the technical optimizations in the Intel Data Platform to other big data use cases throughout the enterprise.

For more information on Intel IT best practices, visit www.intel.com/IT.

## IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation:

- Twitter
- #IntelIT
- LinkedIn
- IT Center Community

Visit us today at **intel.com/IT** or contact your local Intel representative if you would like to learn more.

## Related Content

Visit **intel.com/IT** to find content on related topics:

- Intel IT Best Practices for Implementing Apache Hadoop* Software paper
- Integrating Apache Hadoop* into Intel's Big Data Environment paper
- Mining Big Data in the Enterprise for Better Business Intelligence paper