2016 3rd Asia-Pacific World Congress on Computer Science and Engineering

Architecture for Complex Event Processing using Open Source Technologies

Sanjay Jha
CQUniversity
Sydney, NSW 2000, Australia
s.jha@cqu.edu.au

Meena Jha
CQUniversity
Sydney, NSW 2000, Australia
m.jha@cqu.edu.au

Liam O'Brien
Dept of Immigration and Border Protection, Canberra, ACT 2609, Australia
liamob99@hotmail.com

Pramod Kumar Singh
Hewlett Packard Enterprise, Sydney, NSW-2000, Australia
pramod.singh5@hpe.com

***Abstract*— Big Data, more than ever, is playing a vital role in IT decision making, with such decisions increasingly moving towards being made in real-time. Organisations are optimizing service performance, better handling capacity across overall organisations, and effectively making decisions utilising operational analytics. Realizing the full value of business data is a key challenge for today's operational analytics. Complex Event Processing (CEP) is a technique for tracking, analyzing, and processing data as an event happens and is useful for Big Data because it is intended to manage data in motion. Data in motion is processed and communicated based on business rules and processes. For decisions to be better-informed, data used for decision making has to be timely, complete, accurate, trusted, valid, reliable, and relevant. CEP utilizes data generated from moment-to-moment from different emerging sources such as sensor, sentiment, geo-locational, etc… There is a need to bridge the gap between traditional business intelligence with new Big Data technologies such as CEP. Bridging of this gap will enable organisations to become agile and data-driven so that business outcomes can be maximized by delivering better-informed decisions about a customer and delivering a better service to them. In this paper we discuss the architecture developed for CEP using open source technologies and show how CEP is applied to the use case of an Electronic Coupon Distribution Service (ECDS), using location information, past shopping/travel history, gender, likes/dislikes, etc… We further explore how different types of data such as static information (gender, age, etc.), previous history (where the person travelled to, what they bought, etc.), as well as real-time information about a customer (current location, current shopping habits, etc.) would all be utilised in CEP.**

Keywords*— *Big Data; Complex Event Processing; Business Intelligence; Electronic Coupon Distribution Service; Conventional Coupon Distribution Service.

I. INTRODUCTION

Business intelligence is about providing a clear understanding of and access to the information needed to make decisions at the right time, in the right place, and in the right format. To achieve this, an organization needs a clear articulation of what decisions are being made within the organization, what information is supporting these decisions, where decisions are being made, who the decision makers are, and the applications that support these decisions using data and information [1]. Traditionally in most organizations Information Technology (IT) departments have been slow to support analytics on IT-hosted platforms connected to IT-hosted data. Most organizations are sticking with their Enterprise Resource Planning (ERP) systems for now. But as the number of data sources rises, many organizations are having trouble integrating ERP systems with other data or systems. Data and information collected for decision making needs to be relevant, timely and for the right users at the right time in the right format to drive business value based decisions. Organizations need to get full value from the massive amounts of information they already have within their organizations. New technologies are collecting more data than ever before, yet many organizations are still looking for better ways to obtain value from their data and compete in the marketplace [2]. Using data to make decisions adds value to business decisions. The ability to organize communities of Web participants to develop, market, and support products and services has moved from the margins of business practice to business value based decisions [3]. The data and information explosion is leading to an unprecedented ability to store, manage and analyze data. With diversified data provisions, such as sensor networks, telescopes, scientific experiments, and high throughput instruments, the datasets increase at an exponential rate [4].

Information and data are central components of our everyday activities. Social networks, smart portable devices, intelligent cars, the Internet of Things, smart cities, and smart business transactions represent a few instances of a pervasive information-driven vision we call Big Data. Business Intelligence and Data Analytics and the related field of Big Data analytics, that analyses this information and data, have become increasingly important in both the academic and the business communities over the past decade. Emerging academic research suggests that organizations that use Big Data and business analytics to guide decision making are more productive and experience higher returns on equity than competitors that don't [5].

Complex Event Processing (CEP) has evolved into the paradigm of choice for the development of monitoring and reactive applications [6]. It also has a strong impact on information systems and the way information is subscribed and consumed [6]. CEP enables real-time analysis by utilizing stream data generated from moment-to-moment to

978-1-5090-5753-5/16 $31.00 © 2016 IEEE
DOI 10.1109/APWC.on.CSE.2016.42

support better insight and decision making. With the recent explosion in data volume, variety, velocity, and diversity of data sources, this goal can be quite challenging for architects to achieve.

CEP is a type of event processing that combines data from multiple sources to identify patterns and complex relationships across various events. The value of CEP is that it helps identify opportunities and threats across many data sources and provides real-time alerts to act on them. Today, CEP is used across many industries in a variety of use cases, including:

- Finance: Trade analysis, fraud detection
- Airlines: Operations monitoring
- Healthcare: Claims processing, patient monitoring
- Energy and Telecommunications: Outage detection.

CEP was developed at Stanford University in the mid-1990s by Professor David Luckham [7]. The goal of CEP is to enable information contained in the events flowing through all of the layers of the enterprise IT infrastructure to be discovered, understood in terms of its impact on high-level management goals and business processes, and acted upon in real-time to make well-informed decision. CEP implementations are built around events such as a new purchase, a change of address, or an attempt to break into a network. Events can come from people, devices, applications, networks, or databases. Events can generate responses, or actions. For example, an “Attempted Fraud” event may trigger, in some cases, a “Put Account on Referral” action to make sure downstream account activity is legitimate. The need of CEP represents a paradigm shift in the approach to understanding and responding to business activity to make well-informed decision through IT infrastructure.

SASE [8] an event processing system that executes complex event queries over real-time streams of RFID readings extending existing event languages to meet the needs of RFID-enabled monitoring applications has been proposed by Wu, Diaoa and Rizvi. Agrawal, et al. [9] present an evaluation model and query evaluation framework for pattern matching over CEP that allows for optimization and techniques to improve runtime efficiency. The Cayuga System [10] was built at Cornell as a high-performance system for complex event processing. Cayuga is based on nondeterministic finite automata with buffers. Bry and Eckert in [11] claim that a sufficiently expressive language must be able to handle data (event object property) extraction, event composition (matching multiple events), temporal and causal relationships, and event accumulation (e.g., for aggregation of data over time, or checks for missing events, e.g., order not filled in a specific time). Soberg et al. [12] devise a CEP system that detects deviations from expected events. The paper describes a query language, which is only interesting in the operators it provides to detect deviations. Unfortunately none of the identified work utilizes real-time analytics for decision making or use open source software to build CEP.Real-time systems need to perform analytics on short time windows, i.e. correlating and predicting events streams generated over the last few minutes, which is different to batch processing systems. However, to make decisions these two types of systems, real-time and batch processing, need to be combined so that decisions are well-informed. For instance, a credit card fraud prediction system could leverage a system using previous credit card transaction data over a period of time. This can be combined with a real-time system to find if there is any deviation in the real-time stream data. If a deviation is beyond a certain threshold, it can be tagged as an anomaly. There are number of open source technologies available to process data in batch form and in real-time. Combining these two will create an architecture for CEP. Agile Predictive Analytics is established on a set of core values and guiding principles. It is not a rigid or prescriptive methodology; rather it is a style of building business intelligence applications, and analytics applications that focuses on the early and continuous delivery of business value [18]. The Manifesto for Agile Predictive Analytics is based on: individuals and interactions over process and tools; End-user and stakeholder collaboration over contract negotiation; and Responding to change over following a plan.

In a survey report by Bloomberg Businessweek [19], 97% of companies with revenues exceeding $100 million were found to use business analytics. A report by Manyika et al. [20] predicted that by 2018, the United States alone will face a shortage of 140,000 to 190,000 people with deep analytical skills, as well as a shortfall of 1.5 million data-savvy managers with the know-how to analyse Big Data to make effective decisions.

In this paper we discuss the architecture developed for a CEP system using open source technologies and show how CEP is applied to the use case of an Electronic Coupon Distribution Service, using location information, past shopping/travel history, gender, likes/dislikes, etc… We further explore how different types of data such as static information (gender, age, etc.), previous history (where the person travelled to, what they bought, etc.), as well as real-time information about a customer (current location, current shopping habits, etc.) would all be utilised in CEP. In this paper we will show how real-time analytics use cases can be solved using popular open source technologies to process real-time data.

The remainder of the paper is structured as follows. Section II describes open source technologies for CEP. Section III describes an activity diagram for Electronic Coupon Distribution Service. Section IV outlines an architecture for Complex Event Processing using Open Source Technologies. Finally Section V concludes the paper.

II. OPEN SOURCE TECHNOLOGIES FOR CEP

A. Hadoop

One of the key technologies is Hadoop. Hadoop is a batch processing system and has evolved and matured over the past few years for offline data processing platform for Big Data. Hadoop is a whole ecosystem of technologies designed for the storing, processing and analysing data. The core Hadoop technologies work on the principle of breaking up and distributing data into parts and analysing those parts concurrently, rather than tackling one monolithic block of data all in one go. It is more efficient to break up and distribute data into many parts, allowing processing and analyzing of different parts concurrently. The main advantages of Hadoop are its cost and time effectiveness. Cost, because as it is open source, it is free and available for anyone to use, and can run off cheap commodity hardware. Time, because it processes multiple parts of the data set concurrently, making it a comparatively fast tool for in-depth analysis.

- The Apache Software Foundation [13] are constantly updating and developing the Hadoop ecosystem such as Hadoop on Premium. Hadoop on Premium services such as Cloudera, Hortonworks and Splice offer the Hadoop framework with greater security and support, with added system and data management tools and enterprise capabilities. Some key components of Hadoop include:
- HDFS: Hadoop Distributed File System which is the default storage layer. HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers. HDFS has demonstrated production scalability of up to 200 PB of storage and a single cluster of 4500 servers, supporting close to a billion files and blocks. HDFS requires minimal operator intervention, allowing a single operator to maintain a cluster of 1000s of nodes [14].
- MapReduce: This is composed of two components Map and Reduce. The Map job distributes a query to different nodes, and the Reduce gathers the results and resolves them into a single value. MapReduce is composed of several components such as: JobTracker which is the master node that manages all jobs and resources in a cluster, TaskTrackers which are agents deployed to each machine in the cluster to run the map and reduce tasks, and JobHistoryServer which is a component that tracks completed jobs, and is typically deployed as a separate function or with JobTracker
- Not Only SQL (NoSQL): NoSQL is involved in processing large volumes of multi structured data. Most NoSQL databases are most adept at handling discrete data stored among multi structured data. Some NoSQL databases, like HBase, can work concurrently with Hadoop.

NoSQL is better suited for operational tasks, interactive workloads based on selective criteria where data can be processed in near real-time. Hadoop is better suited to high-throughput, and in depth analysis. Hadoop and NoSQL products are sometimes marketed concurrently. Some big names in NoSQL field include Apache Cassandra, MongoDB, and Oracle NoSQL. Many of the most widely used NoSQL technologies are open source, meaning security and troubleshooting may be an issue. It also places less focus on atomicity and consistency than on performance and scalability. Premium packages of NoSQL databases (such as Datastax for Cassandra) work to address these issues.

Massively Parallel Processing (MPP) Databases work by segmenting data across multiple nodes, and processing these segments of data in parallel. Whereas Hadoop usually runs on cheaper clusters of commodity servers (allowing for inexpensive horizontal scale out), most MPP databases run on expensive specialized hardware (data warehouse appliances). MPP technologies process massive amounts of data in parallel. It may have hundreds (or potentially even thousands) of processors, each with their own operating system and memory, working on different parts of the same programme. MPP uses SQL, and Hadoop uses Java as default (although the Apache Foundation developed Hive, a language used in Hadoop similar to SQL, to make using Hadoop slightly easier and less specialist). Many of the major players in the MPP market have been acquired by technology vendors. Netezza, for instance is owned by IBM, Vertica is owned by HP and Greenplum is owned by EMC. Hadoop is a high-throughput system which can crunch a huge volume of data using a distributed parallel processing paradigm called MapReduce. But there are many use cases across various domains which require real-time / near real-time response on Big Data for faster decision making. Hadoop can be used for building a prediction model for sequence analysis with the help of the Machine Learning library.

B. *Apache Spark Core Engine*

Traditional data warehouses have focused on support for strategic Business Intelligence (BI). In operational data warehousing, the closer the warehouse is to real-time information, the more actionable it becomes for front line users. CEP engines are utilized for rapid and large-scale data processing in real time.

One open source CEP solution is the Apache Spark framework. Apache Spark is used on top of HDFS and promises speeds up to 100 times faster than the two step MapReduce function. This allows data to be loaded in memory and queried repeatedly; making it suitable for machine learning algorithms. An increase in performance is obtained by leveraging computations in-memory. Apache Spark is a fast and general engine for large-scale data processing. Apache Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3. Apache Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming as shown in Figure 1. These libraries can be combined seamlessly in the same application. Apache Spark can be used interactively from the Scala, Python and R shells. Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.

Spark is being adopted by several companies in the industry. To mention a few, Guavus has built its operational intelligence platform on Spark [15], Zoomdata [16] is using SparkSQL to do business intelligence-style analytics and

Graphflow [17] has used Spark to build a real-time recommendation and customer intelligence platform. The Spark engine is a multi-faceted tool that provides a suite of packages to build a variety (online streaming, batch processing, machine learning, etc.) of applications.

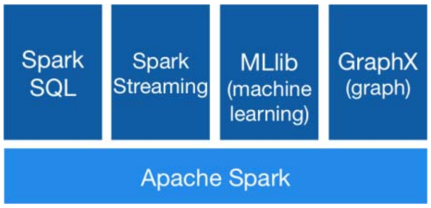

Figure 1: Apache Spark and stack of libraries

Image used from http://Spark.apache.com

C. Spark APIs: Scala or Python for Spark

The Spark provides an API for distributed data analysis and processing in four different languages: Scala, Java, Python, R and DataFrame. Java is complex to learn and quite verbose. Java does not support Read-Evaluate-Print Loop (REPL) interactive shell. Python is a general purpose programming language with excellent libraries for data analysis like Pandas and scikit-learn. But like R, it's still limited to working with an amount of data that can fit on one machine. This is easy to learn. Scala is less complex than Java but more complex than Python. Scala presents a learning curve. But at least, any Java library can be used from within Scala. R offers a rich environment for statistical analysis and machine learning, but it has some rough edges when performing many of the data processing and cleanup tasks before the real analysis work. DataFrame makes Spark programs more concise and easier to understand, and at the same time exposes more application semantics to the engine. A comparative study of Scala and Python based on some attributes such as performance, Learning curve, Ease of Use, Libraries, and Porting R Code is shown in Table 1.

Spark is built on Scala, thus being proficient in Scala helps digging into the source code when something does not work as expected. When Python wrapper calls the underlying Spark codes written in Scala running on a JVM, translation between two different environments and languages might be the source of more bugs and issues. Since Spark is implemented in Scala, using Scala allows accessing the latest greatest features. Most features are first available on Scala and then port to Python. Scala is designed for distributed systems. Hence performance is better than with traditional languages like Python and R. Scala is being integrated well with the big data eco-system, which is mostly JVM based.

CEP challenges are very complex. Big Data is commonly categorized into volume, velocity, and variety of the data, and Hadoop like system handles the Volume and Varity part of it. Along with the volume and variety, the CEP system needs to handle the velocity of the data as well. And handling the velocity of Big Data is not an easy task.

Attributes	***Scala***	***Python***
Performance	+ (uses Java Virtual Machine)	- (slower than C)
Learning Curve	- (java alternative)	+(large community support and tutorials)
Ease of Use	- (Complex to learn)	+ (grand library collection from community)
Libraries	- (small libraries for Machine Learning)	+ (better libraries in Machine Learning and Natural Language Processing)
Porting R code	- (complex way for calling R routines)	+ (easy ways to call R directly from Python)
Type of systems used in	Good for large scale systems	Good for simple to moderately complex analysis and for a quick demo
Typed Language	Scala is statically typed. But looks like dynamic-typed language because it uses a sophisticated type inference mechanism.	Python is dynamically typed
Spark Streaming Data	Scala is preferred	Python supports for Spark streaming but only for basic sources like text files and text data over sockets.
Kind of programming Language	Scala is multi-paradigm programming language that promotes usage of functional principles.	Python is an interpreted language hence slow.

Table 1: Comparisons of Scala and Python for Spark

D. Apache Kafka

Kafka is a tool that is built to handle ingesting transaction logs and other real-time data feeds. Kafka works well as a replacement for a more traditional message broker. Message brokers are used for a variety of reasons (to decouple processing from data producers, to buffer unprocessed messages, etc). In comparison to most messaging systems Kafka has better throughput, built-in partitioning, replication, and fault-tolerance which makes it a good solution for large scale message processing applications.

The original use case for Kafka was to be able to rebuild a user activity tracking pipeline as a set of real-time publish-subscribe feeds. This means site activity (page views, searches, or other actions users may take) is published to central topics with one topic per activity type. These feeds are available for subscription for a range of use cases including real-time processing, real-time monitoring, and loading into Hadoop or offline data warehousing systems for offline processing and reporting. Kafka is often used for operation monitoring data pipelines. This involves aggregating statistics from distributed applications to produce centralized feeds of operational data.

Many people use Kafka as a replacement for a log aggregation solution. Log aggregation typically collects physical log files off servers and puts them in a central place (a file server or HDFS perhaps) for processing. Kafka abstracts away the details of files and gives a cleaner abstraction of log or event data as a stream of messages. This allows for lower-latency processing and easier support for multiple data sources and distributed data consumption. In comparison to log-centric systems like Scribe or Flume, Kafka offers equally good performance, stronger durability guarantees due to replication, and much lower end-to-end latency.

III. ELECTRONIC COUPON DISTRIBUTION SERVICE WORKFLOW MODEL

Luckham [7] gives the following steps for a design of a CEP system:

- Design a new process.
- Convey design to stakeholders; form consensus.
- Simulate on expected data; update design.
- Integrate into system, test; update design.
- Monitor upgraded system.
- Modify system based on monitored results or business requirements.

Electronic coupons are one of the ways to raise popularity for a service, product or brand. Electronic coupons (or special offers, as most vendors prefer to call them) are generally embraced by customers, regardless of financial potential. One of the challenges of Electronic Coupon Distribution Service (ECDS) using locational information is distribution of individually tailored coupon and promotion, with real time analysis of customer information and present location. Customers are moving with mobile devices and looking for certain type of product or store. If the ECDS delivers the coupon to the customers on their mobile device at right time, and at right location, it will improve the ability to attract customers in shops. The objectives of implementing ECDS are to have highest customer satisfaction rating and improvement of the ability to attract customers in shops to increase sale and hence profit for the organisation. Figure 2 depicts a workflow model of ECDS.

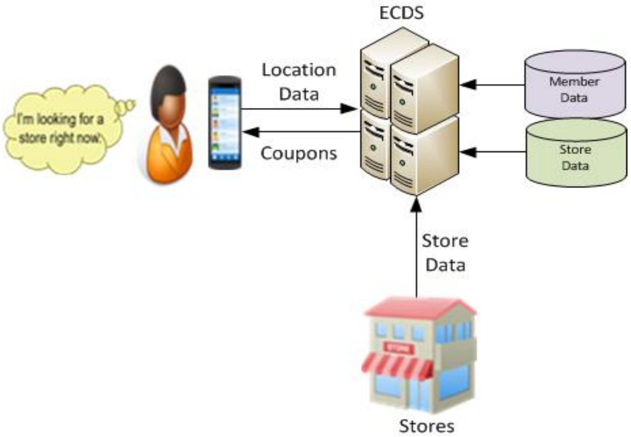

Figure 2: Workflow model of ECDS

To accomplish the workflow model of ECDS the components required to construct the architecture to support ECDS are as follows:

- Sensing Devices such as mobile, Facebook, Twitter, creating structured and unstructured data
- Data Platform service for storage of structured data, CEP, Hadoop.
- Navigation to internal or external Application hosted on cloud.
- End-customers and customers systems where the coupons will be delivered.

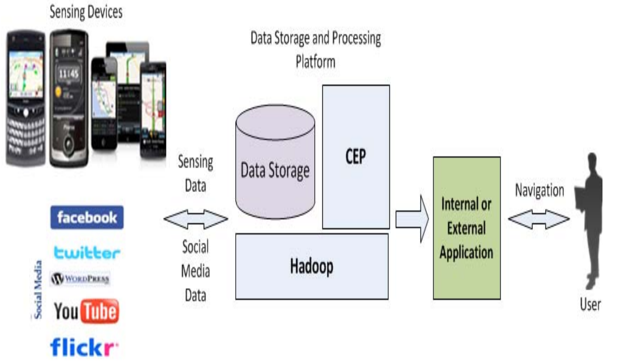

Figure 3: Architectural Components

The components of the architecture are shown in Figure 3. Sensor sends the locational information to CEP. High performance filter is being used to filter the information for processing. CEP takes Member information and Store information from storage of structured data and Hadoop and processes the data with some additional rules to make decisions and then recommendations are being navigated to end users and customer systems.

IV. ARCHITECTURE FOR CEP FOR ECDS

Coupons are certificates that entitle the bearer to stated savings on the purchase of a specific product or product bundle. Conventionally manufacturers and merchants distribute coupons via newspaper inserts, in magazines, or by direct mail. To get the rebates using coupons usually require the customer to redeem the rebate certificate by mailing it to the manufacturer along with proof of purchase. Some of the popular uses for coupons include: promoting a new brand (manufacturer-issued coupons); persuade customers to switch to the promoted brand (manufacturer-issued coupons); increase sales of an existing product (both manufacturer- and merchant-issued coupons) and attract shoppers to a retail establishment (merchant-issued coupons).

The conventional approach to distribute coupons is to issue identical coupons regularly to all customers as shown in figure 4. The conventional approach to distribute coupons faces many disadvantages such as:

- conventional distribution systems are slow and have long lead-times;
- coupons do not get to the targeted customers;
- third, redemption rates are very low, 1% of the distributed coupons [21].

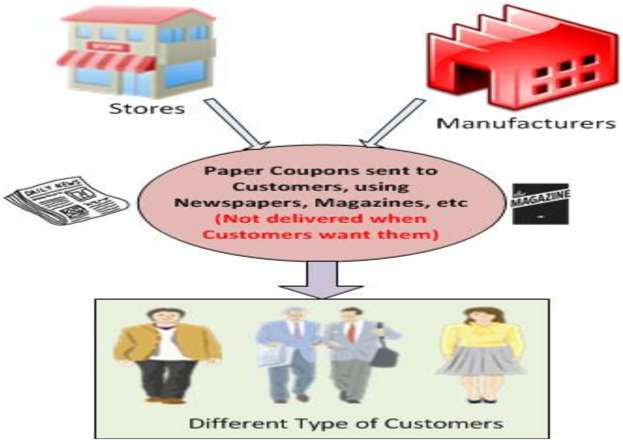
Stores
Manufacturers
Paper Coupons sent to Customers, using Newspapers, Magazines, etc
(Not delivered when Customers want them)
Different Type of Customers

Figure 4: Conventional way of Coupon Distribution

The coupon concept has not been adopted widely on the Internet. Several Web sites offer printable versions of conventional coupons, these coupons cannot be redeemed online [22]. E-coupon issuers enjoy a high degree of flexibility in choosing which e-coupons are given to shoppers and when they are offered. For example, e-coupons could be offered to shoppers when they enter an online store, when they view a product description, or when they finalize their purchases. Similarly, e-coupons could be offered for a product for which a shopper has expressed interest, a product related to the product a shopper is buying, or a product the shopper never buys but the storekeeper is interested in promoting.

Architecture for CEP requires capturing clickstream. A clickstream is recording of the parts of the screen a computer user/ mobile user clicks on while web browsing as shown in Figure 5.

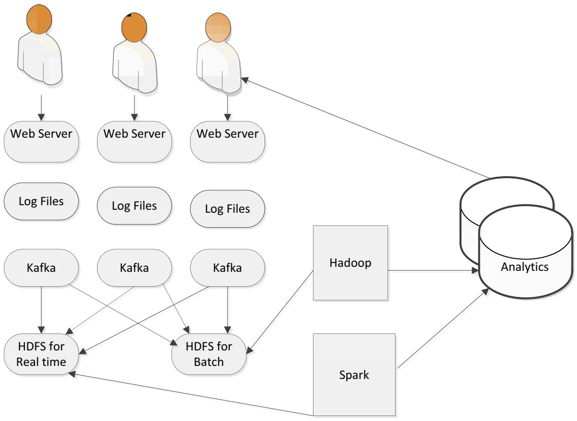

Figure 5: Architecture of CEP for Coupon Distribution

Coupons are sent based on location information, store status, previous history and profile of the customer, gender, age, previous history what they bought and real time information about a customer. To send the coupon to interested customers require to process, locational data, members and store information data. Members and store information requires filtering large volume of location data which is being generated by log files. Locational information (Real-time event data) is being sensed by the sensing devices such as mobile, and member and store information is stored in the database. These two information needs to be combined with the rules to develop recommendations and make decisions for the customers as shown in Figure 6.

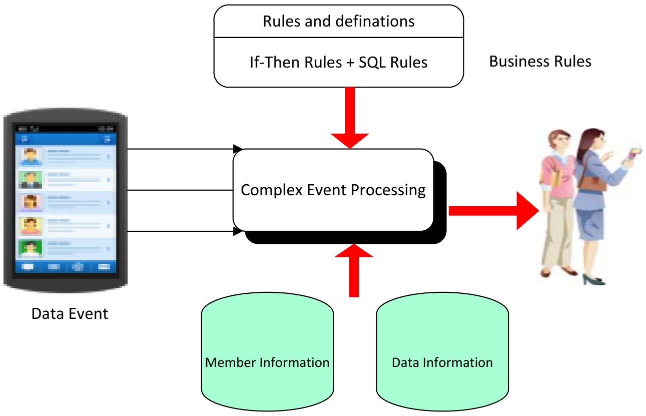

Figure 6: Information Processing using CEP

Business rules are frequently defined based on the occurrence of scenarios triggered by events [23]. Rules must be set in advance in order to execute complex event processing. Examples are such as: take an action if the customer is identified as a loyal customer based on his purchase habits. Business rules and event processing queries change frequently and require immediate response for the business to adapt itself to new market conditions, new regulations and new enterprise policies. Business rules can be divided into two types: One is If-Then-Else and the second one is SQL rules.The rules are based on three components and they are:

- ***Event:*** defines the sources that can be considered as event generators such as sensors;
- ***Condition:*** specifies when an event must be taken into account; for example, we can be interested in some data only if it exceeds a predefined limit;
- ***Action:*** identifies the set of tasks that should be executed as a response to an event detection: some systems only allow the modification of the internal database, while others allow the application to be notified about the identified situation.

To execute the rules at runtime, five phases have been identified [24]. These are as follows:

- ***Signalling***: detection of an event;
- ***Triggering***: association of an event with the set of rules defined for it;
- ***Evaluation*****:** evaluation of the conditional part for each triggered rule;
- ***Scheduling***: definition of an execution order between selected rules;
- ***Execution*****:** execution of all the actions associated to selected rules.

CEP puts great emphasis on the issue and ability to detect complex patterns of incoming data involving sequencing and ordering relationships. CEP relies on the ability to specify composite events through event patterns that match incoming event notifications on the basis of their content and on some ordering relationships on them. CEP requires interaction with a large number of distributed and heterogeneous information data sources and sinks which observe the external world and operate on it. This is typical of most CEP scenarios, such as environmental monitoring, business process automation, and control systems.

V. RESULTS ACHIEVED BY USING ECDS

Information is going to be our generation's next natural resource like steam was to the 19th century [25]. Mobile is everywhere – more people have a cell phone than running water and 25% of the world will be on a social network – that's what has created big data: 2.5 billion gigabytes of data is created per day [25]. The organization like Macy's and Kohl's switched from sending shoppers blanket email promotions to sending targeted offers based on individual shopper purchases [25].

The organization using ECDS recorded their sale and customer satisfaction data and has analysed the results based on the attributes such as: Use of coupons; receiving discount coupon on time; customer satisfaction and improvement of the ability to attract customers in shops/ customer loyalty. The results achieved by using ECDS over Conventional coupon distribution system have been summarized in Table 2.

Comparison Features	***Results achieved using ECDS***	***Conventional Coupon Distribution system***
Use of Coupons	46% coupons were used.	5% coupons were used.
Receiving Discount Coupon on Time	Yes, 89% Customers received online coupon on time.	No, only 15% received coupons on time.
Customer Satisfaction	12% increase in customer satisfaction from 50% to 64%.	2% customer rated as satisfied. 50% to 52%.
Customer Loyalty	Retention rate increased to 66% which is 10% higher.	56% customers shop.

Table 2: Result Summary of ECDS over Conventional Coupon Distribution System

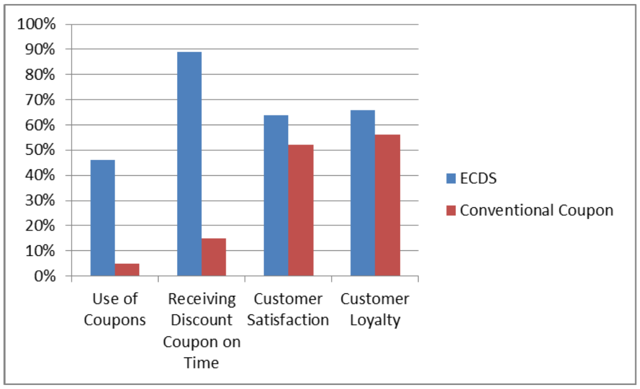

Figure 7: Visualization of the Results

The visualization of the results is shown in Figure 7 displays a very positive response on the need of ECDS for organizations in order to improve customer satisfaction and to improve the ability to attract customers in shops. To send coupons on the right time to the right customers at the right location requires decisions to be made based on the historical data and the locational data of the customers. This was achieved using ECDS where Hadoop is used for batch processing and spark is used for real-time processing.

VI. CONCLUSION AND FUTURE WORK

E-Coupon is one of the ways to attract customers and increase customer satisfaction. Conventional ways of distributing coupons are using newspapers, magazines, emails etc… The conventional process to distribute coupons targets all kind of customers and not differentiating between who will buy and who will not. Sending customers blanket promotions and coupons shows that only 5% of the coupons are used 95% coupons are wasted and not used.

In this paper we have discussed the architecture developed for CEP using open source technologies such as Hadoop, Apache Kafka, Spark and Scala as a language and documented the results showing how CEP is applied to the use case of an ECDS, using location information, past shopping/travel history, gender, likes/dislikes, etc… We have explored different types of data such as static information (gender, age, etc.), previous history (where the person travelled to, what they bought, etc.), as well as real time information about a customer (current location, current shopping habits, etc.) to send coupons to customers. The architecture of ECDS allows customers going online gets connected to web servers generates weblogs. Apache Kafka is a tool that is built to handle ingesting transaction logs and other real-time data feeds. HDFS is used for batch and real-time processing. The batch processing is done using Hadoop and real-time processing is done using Spark and the results are being sent to analytics platform.

With the use of ECDS, it shows that 46% of the distributed coupons are used. Coupons and discounts do help sales, and if used and targeted well, can be effective in driving business forward. Poor coupon redemption is a poor KPI and Big Data can help in improving the redemption of discount coupons. Big Data often helps in discovering the past history of the customer, the shopping habits, the product they looked for and the product they would like to buy.

REFERENCES

[1] H. Chen, R. H. L. Chiamg, V. C. Storey, “Business Intelligence and Analytics: From Big Data to Big Impact”, *MIS Quarterly,* Vol. 36, No. 4, Page(s): 1165-1188, December (2012)

[2] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins and N. Kruschwitz, “Big Data, Analytics and the Path From Insights to Value”, *MIT Sloan Management Review*, Vol 52, No. 2, Winter (2011)

[3] J. Bughin, M. Chui, and J. Manyika, “Clouds, Big Data, and Smart Assets: Ten Tech-enabled Business Trends to Watch”, *McKinsey Quarterly*, (2010)

[4] A. S. Szalay, “Exterme Data-intensive Scientific Computing”, *Journal of Computer Science and Engineering*, Vol.13, No.6, Page(s): 34-41, (2011)

[5] E. Brynjolfsson, L. M. Hitt, and H. H. Kim, “Strength in numbers: How does Data-driven Decision Making Affect Firm Performance? ”, *Social Science Research Network (SSRN)*, April (2011)

[6] A. Buchmann, B. Koldehofe, “Complex Event Processing” *Journal of IT-Information Technology*, Vol.51, No. 5, Page(s):241–242, September (2009)

[7] D. Luckham, “*The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems”*, Pearson Education, Inc. (2002)

[8] E. Wu, Y. Diao, and S. Rizvi, “High-Performance Complex Event Processing over Streams”, *Proceedings of the 2006 ACM International Conference on Special Interest Group on Management of Data (SIGMOD),* June 27-29, Chiocago, Illinios, USA, (2006)

[9] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman, “Efficient Pattern matching Over Event Streams”, *Proceedings of the 2008 ACM International Conference on Management of Data (SIGMOD)*, New York, NY, USA, 147-160, (2008)

[10] L. Brenna, A. Demers, J. Gehrke, M. Hong, J. Ossher, B. Panda, M. Riedewald, M. Thatte, and W. White, “Cayuga: A Highperformance Event Processing Engine”, *Proceedings of the International Conference on Management of Data (SIGMOD '07),* ACM, New York, NY, USA, (2007)

[11] F. Bry and M. Eckert, “Temporal Order Optimizations of Incremental Joins for Composite Event Detection”, *Proceedings of the Inaugural International Conference on Distributed Event-Based Systems (DEBS '07),* ACM, New York, NY, USA, Page(S): 85-90, (2007)

[12] J. Søberg, V. Goebel, and T. Plagemann, “To happen or Not to happen:Towards an Open Distributed Complex Event Processing System”, *Proceedings of the 5th Middleware doctoral symposium (MDS '08)*, ACM, New York, NY, USA, Page(s): 25-30, (2008)

[13] Apache Software Foundations, Accessed from:
http://www.apache.org/

[14] HDFS Java API, Accessed from:
http://hadoop.apache.org/core/docs/current/api/

[15] E. Carr, “Building Big Data Operational Intelligence Platform with Apache Spark”, Guavus, Spark Summit, (2014)

[16] J. Langseth, “BI-style Analytics on Spark (without Shark) using SparkSQL and SchemaRDD”, Zoomdata, Spark Summit, (2014).

[17] N. Pentreath, “Using Spark and Shark to Power a Real-time Recommendation and Customer Intelligence Platform”, Graphflow, Spark Summit, (2014)

[18] K. Collier, “*Agile Analytics: A Value Driven Approach to Business Intelligence and Datwarehusing*”, Pearson Education, ISBN 978-0-321-50481-4, (2012)

[19] Bloomberg Businessweek. “The Current State of Business Analytics: Where Do We Go from Here?”, Bloomberg Business- week Research Services, (2011), Accessed from:
http://www.sas.com/resources/asset/

[20] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” McKinsey Global Institute, (2011)

[21] R. Anand, M. Kumar, A. Jhingran, and R. Mohan, “Sales Promotions on the Internet, “*Proceedings of the Second Usenix Conference on E-Commerce*, Boston, (1998)

[22] P. Kotler and G. Armstrong, “*Principles of Marketing*”, Prentice Hall, (1998)

JBoss, “ Complex Event Processing”,(2009) Chapter 8, Accessed from:
https://docs.jboss.org/.../DroolsComplex**EventProcessing**Chapter.html,

[23] G. Cugola, A. Margara, “Processing Flows of Information: From Data Stream to Complex Event Processing”, *Journal of ACM Computing Surveys (CSUR) Surveys Homepage Archive*, Vol. 44, No.3, June 2012 Article No. 15 ACM New York, NY, USA, (2012)

[24] B. Thau, “ How Big Dta Helps Stores Like Macy’s And Kohl’s Track You Like Never Before”, Forbes/Retail Janu 24, (2014)
Accessed from: http://www.forbes.com/sites/barbarathau/2014/01/24/