IT@Intel White Paper

Intel IT
IT Best Practices
Big Data Predictive Analytics
December 2013

Reducing Client Incidents through Big Data Predictive Analytics

Ajay Chandramouly
Big Data Domain Owner, Intel IT

Ravindra Narkhede
Enterprise SM Architect, Intel IT

Vijay Mungara
IT Big Data Architect, Intel IT

Guillermo Rueda
IT Big Data Architect, Intel IT

Asoka Diggs
Enterprise Architect, Intel IT

Our new ability to proactively, rather than reactively, identify client issues and implement fixes before they become widespread promises to deliver significant cost avoidance to the enterprise.

Executive Overview

In a proof of concept (PoC) applying natural language processing and statistical modeling to PC client event logs and IT Help Desk incident reports, Intel IT predicted 20 percent of the incidents that occurred in the following 28 days. Our new ability to proactively, rather than reactively, identify and solve potential client issues before they become widespread promises to deliver significant cost avoidance to the enterprise.

In 2013 Intel IT set a target to reduce all reported IT incidents (on clients, servers, and other devices) requiring our attention by 40 percent by the end of the year. Recognizing clients as the primary contributors to overall incidents, we devised a client incident prediction PoC using Intel® Distribution for Apache Hadoop* software (using Hadoop version 2.2). Applying text analytics to millions of client event logs and thousands of client incident reports, we identified correlations enabling us to anticipate and solve client problems before they become widespread.

In performing the PoC, we realized a number of accomplishments.

- Developed a big data predictive analytics solution capable of deriving value from the millions of previously rarely used Windows* event records generated daily by 95,000+ client systems
- Applied advanced natural language processing and information retrieval techniques that enabled correlation of machine information (event data) with internal customer information (incident reports)
- Sorted through millions of events and thousands of incidents, achieving 78-percent accuracy in predicting the occurrence of incidents in additional clients
- Created data visualizations that helped IT support staff quickly determine the likelihood, severity, and distribution of a problem and more accurately target fixes and other proactive support

Combining data mining and predictive analytics, our client incident prediction solution makes it possible for us to find value in data that was once largely ignored. This new capability will enable us to solve many client issues before they have an impact on user productivity. Elements of this solution may prove promising for finding new value in other data logs, such as those collected in Intel's manufacturing, supply chain, marketing, market research, and other operations.

Contents

IT@INTEL

The IT@Intel program connects IT professionals around the world with their peers inside our organization – sharing lessons learned, methods and strategies. Our goal is simple: Share Intel IT best practices that create business value and make IT a competitive advantage. Visit us today at www.intel.com/IT or contact your local Intel representative if you'd like to learn more.

BACKGROUND

Intel's worldwide computing environment includes more than 95,000 enterprise PC clients. A problem reported in one client often will soon appear in hundreds or thousands of similar clients, as well as potentially other IT devices. Maximizing client stability is a high priority since system and application errors and crashes have a serious impact on user productivity, cause loss of unsaved work, and add substantial costs to IT operations. Proactively solving client issues, a long-time goal in IT support, is a key component in our goal to reduce all IT device incidents requiring our attention by 40 percent in 2013.

We define client "incidents" as issues such as system and application errors, malware infections, and other problems that disrupt normal use. Client systems account for approximately 80 percent of total enterprise incidents, making them the primary contributor by volume to overall incidents and a high priority target for IT cost savings.

IT organizations have a wealth of information available to help them identify and find the root cause of many client incidents. The primary information sources include customers reporting incidents in help requests and event log data generated by client machines. "Events" are issues recorded by the Windows* event log system. These logs record events for everything from the failure to start a component to the failure to complete an action. Events are categorized as critical, error, information, and audit.

To use a medical analogy, incidents are like specific patient complaints and daily event logs are like comprehensive lab reports detailing everything potentially wrong in a blood sample. An accurate diagnosis generally requires both. The challenge to proactively analyzing event logs is that event log data can amount to approximately 2,000 logs per day per machine with an average of 40 critical events. Multiplied by 95,000+ clients, this data can total up to 19 million events per day and up to 300 gigabytes of data across the enterprise in a quarter. Making accurate predictions requires more than a day's worth of data. We consider one year of data amounting to 1 terabyte to be ideal.

Prior to the client-incident-prediction proof of concept (PoC), the volume of event log information was overwhelming. Most of it went unanalyzed and unused. Support staff accessed a client's event log only for the short period in which a specific client incident occurred. Even these reactive efforts to relate event log information to an employee help request for a specific client required a great deal of staff time and effort. Trying to keep up with this workload has led many IT organizations to give up the long-term goal of developing the capacity to predict incidents before they happen.

In 2009, Intel IT made major progress on moving from a reactive to a proactive approach to problem management. We developed a tool that collects "blue screen" system crash data from thousands of clients and parses the data to identify the root cause. Through this effort, we were able to categorize and prioritize issues. By deploying solutions for the top-priority

issues, we reduced the number of blue screens from 5,500 a week to fewer than 2,500 a week, often identifying client machines that were likely to experience the same problems, allowing us to fix them before a failure occurred.

Looking to build on our success, we set a goal for our PoC to enable IT support to see and deal with any given incident only once. We wanted to develop a solution capable of matching client incidents with event logs across similar machines and then performing analysis to identify the problem's source. The solution could then be implemented across the enterprise, preventing any recurrence of the problem.

PROOF OF CONCEPT

To increase Intel IT's proactive problem management capabilities for client systems, Intel IT conducted a PoC focused on using the massive amounts of client event log data being largely ignored. Using a big data platform based on Intel® Distribution for Apache Hadoop* software (Intel® Distribution), which uses Hadoop 2.2, we applied text analytics to millions of client event logs and client incident reports. By comparing the results using a data visualization solution, we identified correlations that helped us trace client problems back to their first appearance in the environment. We discovered that we could then anticipate those same problems for other users before those users ever knew the problems existed and, in many cases, apply fixes before the problem appeared.

This section begins by describing the solution architecture and primary tools. It then discusses the collection and analysis of the data, our solution for making the vast amounts of data manageable, and our implementation of visual analytics techniques using a commercially available tool. A key concept referred to in various sections is a grouping construct we used called *symptom*. Described more fully in its own section, a symptom is a group of client events that are essentially identical to each other except that they occurred at different times and on different machines.

Looking to build on our success, we set a goal for our PoC to enable IT support to see and deal with any given incident only once.

Incident Predictability Solution Architecture

The three "Vs"—volume, variety, and velocity—drove our decision to use a big data solution. The three Vs are the defining properties of big data. In the case of our PoC, *volume* refers to the amount of data we needed to process—more than 200 million rows of event data collected over five months. *Variety* refers to the merging of unstructured text data from incidents with event data in XML format—a mixture that required parsing to make the correlations and the resulting IT insights possible. *Velocity* refers to the increasing number of events logged on a continuing daily basis. It also references the speed at which support staff must analyze and merge event and incident data to make a diagnosis and take action to prevent the incident from occurring in other clients.

Problems in Traditional Client Problem Management

Client problem management includes diagnosing the root causes of incidents and determining the resolution of those causes to stop further incidents. It has two primary goals: prevent incidents from recurring and minimize the impact of incidents that cannot be prevented.

Traditional client problem management focuses on recognizing incident trends and pursuing opportunities to analyze root causes using the best possible incident data. This activity is reactive, as it requires users (internal customers) to express dissatisfaction by contacting the service desk to communicate a service degradation or a configuration (hardware or software) issue that has a negative impact on their productivity. Users experience destructive incidents; IT support helps remediate the issue. User productivity is diminished until the issue is fixed.

The data recorded about these incidents is typically subjective since it comes from users and support staff. If several similar incidents occur—and are recorded correctly—IT support *may* be able to spot a trend. But this model has a critical shortcoming: it relies entirely on users reporting issues. In our experience, users do not always report issues and may simply put up with system crashes to avoid the time and trouble of dealing with support staff. They might even think they received a failure-prone system but never notify IT of the problem. Such spotty reporting can make it hard to identify trends and determine how widespread a problem may be.

In our Incident Predictability Solution Architecture based on Intel Distribution (see Figure 1), both the incident data collected from our IT Service Management Request system and the event data are moved daily into the platform. The data then undergoes text analytics processing in the Hadoop cluster to reveal similarities between incident and event patterns.

The PoC used a Hadoop cluster of servers. We used MapReduce to process our large data sets using a parallel, distributed algorithm on a Hadoop cluster. The MapReduce process includes a Map procedure, which filters and sorts the data, and a Reduce procedure, which summarizes the data. In our PoC, the Map and Reduce procedures filtered, sorted, and summarized text, yielding results such as frequencies of common words.

Each MapReduce algorithm may be executed in any node of the cluster. Additionally, Hadoop provides a distributed file system, Hadoop Distributed File System (HDFS*), which stores data on the compute nodes. The HDFS delivers high aggregate bandwidth across the clusters for fast processing. Both MapReduce and the HDFS are designed so that the framework automatically resolves any node failures, ensuring high availability.

Primary Tools

One challenge for a project of this scope and complexity was that the technologies used require a combination of knowledge not generally found in one person. To assemble the necessary knowledge, we took a team approach. In addition to expertise in Hadoop and Linux*—the open source operating system used with Intel Distribution—we also recruited team members with expertise in the key technologies described below to help us find solutions to the various limitations we encountered with our tools.

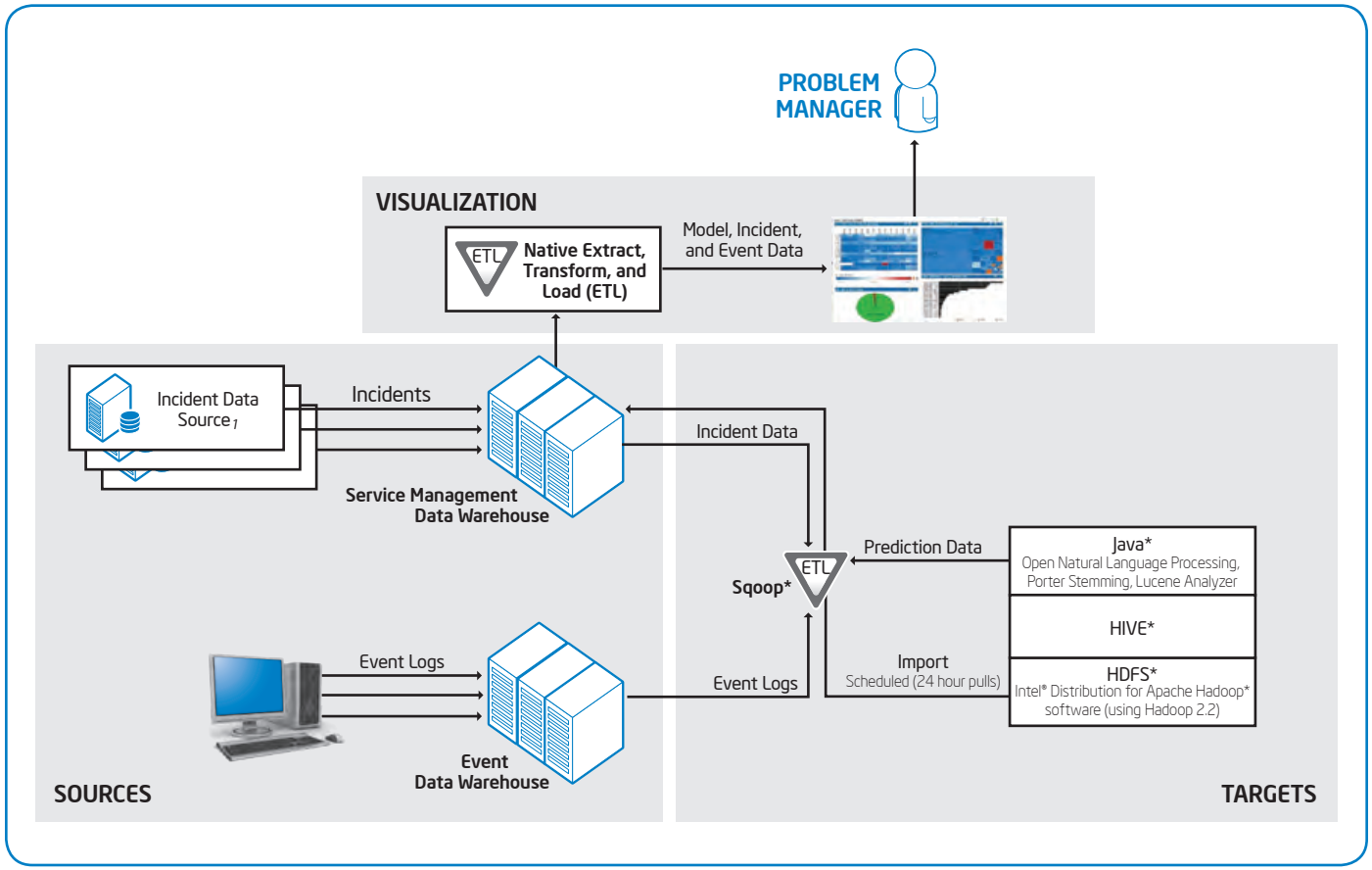

Figure 1. The Incident Predictability Solution Architecture uses the Intel® Distribution for Apache Hadoop* software (using Hadoop 2.2) to process the data and perform text analytics to find similarities between the client event logs and client incident reports.

HIVE*

Hive is a data warehouse infrastructure built on top of Hadoop that provides data summarization, query, and analysis. It features an easy syntax for relational database management system (RDBMS) developers called HiveQL, while maintaining full support for MapReduce algorithms. All the logic used in our PoC was developed using HiveQL except the event-parsing process that was developed using Java* and the text analytics in Java that use the Apache OpenNLP* library.

We faced several challenges with Hive:

- Hive cannot update a table with an update command, so we settled for a two-step update process.
- Because Hive cannot generate identity columns automatically, we implemented a Java function to create a user-defined function for generating a universal unique identifier for symptoms.
- Since Hive does not allow inserting new columns in existing tables, every time we needed to insert a new column we created a new table and moved over the existing data. This two-step procedure added additional processing.

JAVA

Java is a widely used general-purpose, class-based, object-oriented computer language. We used Java for a variety of tasks. The primary use was for events XML parsing: Java read the XML event data and parsed all fields into a Hive table. We also used Java to create the user-defined function for generating universal unique identifiers for symptoms.

We did face some challenges with Java. For instance, after experiencing some of the common memory issues attributed to Java database connectivity, we switched to HDFS for this task. We also experienced Java memory heap space errors when launching a Java virtual machine for a map task. We solved this problem by changing the size settings in the Hadoop job configuration settings (mapred.child.java.opts).

Dependencies in the Java ARchive (JAR) proved challenging as well, particularly, synchronizing our libraries with the Hadoop cluster. We resolved this issue by copying the dependent JARs into the distributed cache of Hadoop, enabling us to synchronize our libraries with all the nodes in our cluster in an efficient way through the HDFS.

APACHE OPENNLP

Apache OpenNLP ("NLP" stands for natural language processing) is a machine learning toolkit for the processing of natural language text. This toolkit supports tokenization, sentence segmentation, part-of-speech tagging, chunking, parsing, and other language processing tasks. In our incident prediction solution, OpenNLP collects inputs for "bag-of-words"—a text analytics technique—by applying NLP techniques to specific events or clients. Through tokenization, stop-words, and stemming, NLP cleans up the unstructured text, preparing it for the further processing that will enable incident prediction.

SQOOP*

Sqoop is a command-line interface application for transferring data between relational databases and Hadoop. In the PoC, we used Sqoop to transfer the historical data in our RDBMS to the Hadoop cluster. One challenge we faced was that Sqoop uses commas as one delimiter in transferring delimited text—text separated into meaningful chunks by specific characters. Since our data includes commas, we used *\t* instead through the hive-delims-replacement option.

Another issue that came up with Sqoop was the appearance of duplicate rows when transferring data for big tables. We resolved this by adding a step for decompressing compressed tables.

DATA VISUALIZATION SOLUTION

Data visualization software provides the necessary treemap, heatmap, and other visualization tools for interacting, exploring, monitoring, and analyzing large data sets. Its exploratory capabilities enable problem managers to drill down to obtain more detail about an outlier or drill up to get an overview that may reveal a trend that was previously hidden. Interactive filters let them remove irrelevant data and noise from their analyses, making underlying patterns easier to see.

Data Collection

Our Incident Predictability PoC used two sets of data:

- **Voice of the customer (VOC)** data, which consists of client issues reported by Intel employees primarily through our IT Help Desk. Employees report issues through this interface by phone, web ticket, fax, or email.
- **Voice of the machine (VOM)** data, which includes all the event logs generated by client machines daily. An event-forwarding system collects this data throughout each day.

Each of these data sets was collected into its own container, which then fed the data into the Incident Predictability Solution architecture designed to correlate the two. The similarity here is that both data sets are text-based. Users describe their problem through the use of words. Event logs consist of errors and warnings that include, along with codes, text-based descriptions that describe what happened on a client system at a particular time with a particular application. Many of these errors and warnings may not be evident to the user. Other errors and warnings may be evident through anything from an application not performing a requested task to a system crash or problem.

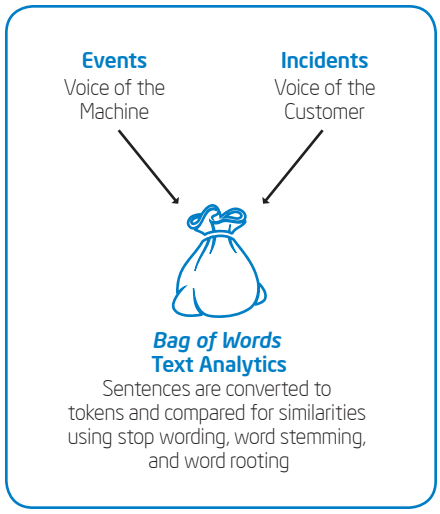

Figure 2. To match events (voice of the machine) with incidents (voice of the customer), the proof of concept applied bag-of-word text analytics to prepare the data in order for us to find similarities.

Big Data Text Analysis

In the past, it would have been impossible to manually sort through and interpret the hundreds of millions of event logs contained in the VOM data. Finding events correlated to a customer-reported incident that also applied to other incidents and could be used to predict future incidents would not have been cost-effective or timely enough to be of value.

With Intel IT's growing use of big data platforms to comb immense amounts of structured and unstructured data for advanced business intelligence for Intel business groups, we began to consider using such a platform for incident prediction. We had looked into traditional solutions, such as a RDBMS, but the dimensional data models on a single query took 32 hours to execute in a RDBMS. We also looked into a cross-platform, document-oriented NoSQL database system. In the end, the sheer amount of data, the data's unstructured nature, and the availability of platforms to run our solution led us to choose an Apache Hadoop solution based on Intel Distribution.

To find connections between the VOC and VOM data involved in certain incidents, we used a bag-of-words model. This simplified technique for processing natural language and retrieving information represents and treats text in a sentence or document as an unordered collection of words. Disregarding grammar and word order, the bag-of-words technique focuses on the existence of a word and may take into consideration criteria such as the frequency of the word's occurrence.

In a bag-of-words analysis, all sentences are first converted to words through tokenization. Tokens are strings of one or more characters—in our case, the letters that form words—that are significant as a group. The quality of the words we keep for analysis is improved through various cleaning techniques, such as the following:

- **Stop wording.** A stop list is used to delete from text various root words that are not relevant to context or not specific enough. Words for removal according to our stop list include articles (such as *a* or *the*), prepositions (such as *of* or *for*), demonstratives (such as *this* or *that*). In addition, some verbs and verb parts are considered insignificant (such as *have*, *can*, *may*, and *would*).
- **Word stemming.** In this step, words are reduced to their stem, or inflectional root, to make it easier to match with other words sharing the same stem. For instance, nouns and verbs sharing the same stem (such as *crash*, *crashes*, *crashed*, and *crashing*)—would be reduced to *crash* for easier matching.
- **Word rooting.** In some cases, instead of stemming, a word simply needs to be reduced to a core root. For example, a word like *failure* can be reduced to its root *fail* and then more easily matched with every instance of the root *fail*.

Before applying the above text analytics techniques, we cleaned the data by searching for known abbreviations of key words in a sentence and replacing them with their full text to improve analysis results. For example, we would replace "IE" with "Internet Explorer."

To match event patterns with client incidents, we applied text analytics on VOM events and VOC information emanating from the same client based on its unique name (see Figure 2).

After completing text analytics, we employed an algorithm based on the Jaccard similarity coefficient—a statistic comparing the similarity and diversity of the two data sets—to find similarities between the processed VOM events and VOC incidents. The algorithm computes the number of words

common to VOM events and VOC incident(s) divided by the total number of words.

The Jaccard similarity coefficient measures the similarity between two sample sets (A and B) by dividing the size of the intersection (in this case, the number of times an event log and an incident report use a common word) by the total of words in both sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The algorithm filters out all pairs with similarity metrics lower than an adjustable threshold to deliver just the best matches: the incidents and associated symptoms on which problem managers will want to perform root-cause analysis. These matches also enable problem managers to predict with a high degree of accuracy which incidents pose the greatest risk of becoming widespread throughout the enterprise.

Making Data Manageable

The incident data used in our PoC came from data collected over a five-year period in our IT Service Management system. The event data came from data collected from the Windows Event Framework over approximately the last six months of incident data collection. The data included 1 million incidents and more than 220 million events.

Events can be grouped by event ID—unique identifiers coded for easy reference by the client's operating system manufacturer. To reduce the number of event IDs to a manageable number, we considered only event IDs that had at least 1,000 instances, resulting in a data set of just over 22 million events. In accordance with the Pareto principle, which states that roughly 80 percent of a set of problems results from 20 percent of the possible causes, we focused on the top 20 percent of event IDs occurring in the environment (see Figure 3). We found that more than 80 percent of the events were grouped in 12 event IDs.

GROUPING EVENTS INTO SYMPTOMS

To reduce problem complexity and enhance root-cause analysis, we used a grouping concept called "symptom." This concept allows us to group events that are essentially the same. A symptom is a group of individual events that are identical to each other, varying only by the time and date the event was recorded and the client on which the event occurred. We based symptom characteristics on XML tags since all event data is in XML format with XML fields containing more than 50 distinct XML tags or columns representing different data elements depending on the characteristics of a specific event ID.

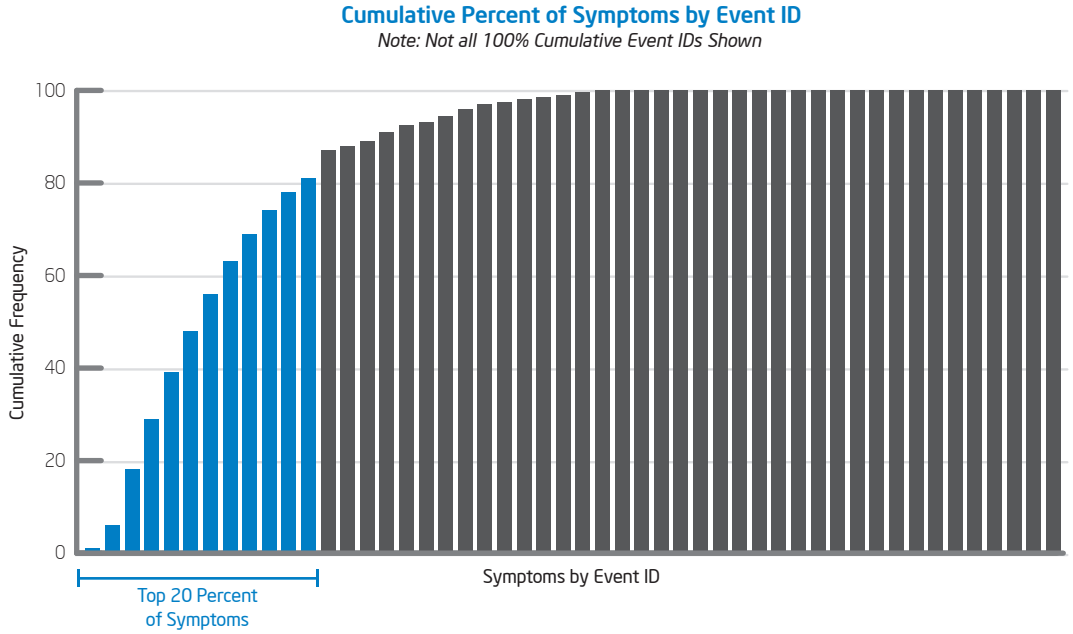

Figure 3. Using the Pareto principle, we concentrated on the top 20 percent of event IDs (blue bars) occurring in the environment. In this cumulative frequency analysis histogram, as we move from left to right, we see the accumulating sum in percentage as the event ID represented in the greatest percentage of events is added to the event ID contributing to the next greatest percentage of events, and so on. The blue bars show that, according to our results, 12 event IDs represent collectively 80 percent of our events. We focused our initial efforts on these 12 event IDs.

To identify symptoms, we used the expertise of our problem managers. For example, for event ID 1000, problem managers translated and helped us define all the critical fields. The particular combination or pattern of values found in those chosen fields constitutes a distinct symptom (see Figure 4).

To find our unique list of symptoms, we used a dozen "select distinct" statements, each of which is specific to one of the 12 event IDs that are responsible for 80 percent of the events we see across the enterprise. Select distinct statements are commands applied to a table that ignore duplicate values and list only the different or distinct values. Through this process we are able to identify new symptoms from new events while adding information to the existing collection of identified symptoms. In our symptoms database, the table is updated on a daily basis.

By applying the symptom concept to our 200 million event rows (five months of events), we were able to group these events into a much more manageable data set of less than 145,000 symptoms.

Once we found the relationship between the VOM from events and VOC from incidents through the text analytics process and the Jaccard similarity metric, we filtered our symptoms for similarities with our incidents using the same metric. This step enabled us to determine the importance of the symptoms in relation to incidents. Problem managers were then able to focus on the symptoms having a high similarity ratio with a high number of incidents.

Visual Data Discovery

Data visualization software is a key element of our solution, providing a fast, accurate way to view and interpret the data. Without this visualization tool, the data would require a statistician to interpret and understand, creating a barrier to hands-on analysis by problem managers. Through treemaps, heatmaps, and other visual representations, our data visualization solution enables problem managers to easily interact with, filter, and understand the data to readily identify the incidents and symptoms that are most likely to contribute to the business objective of proactively identifying recurrent incidents.

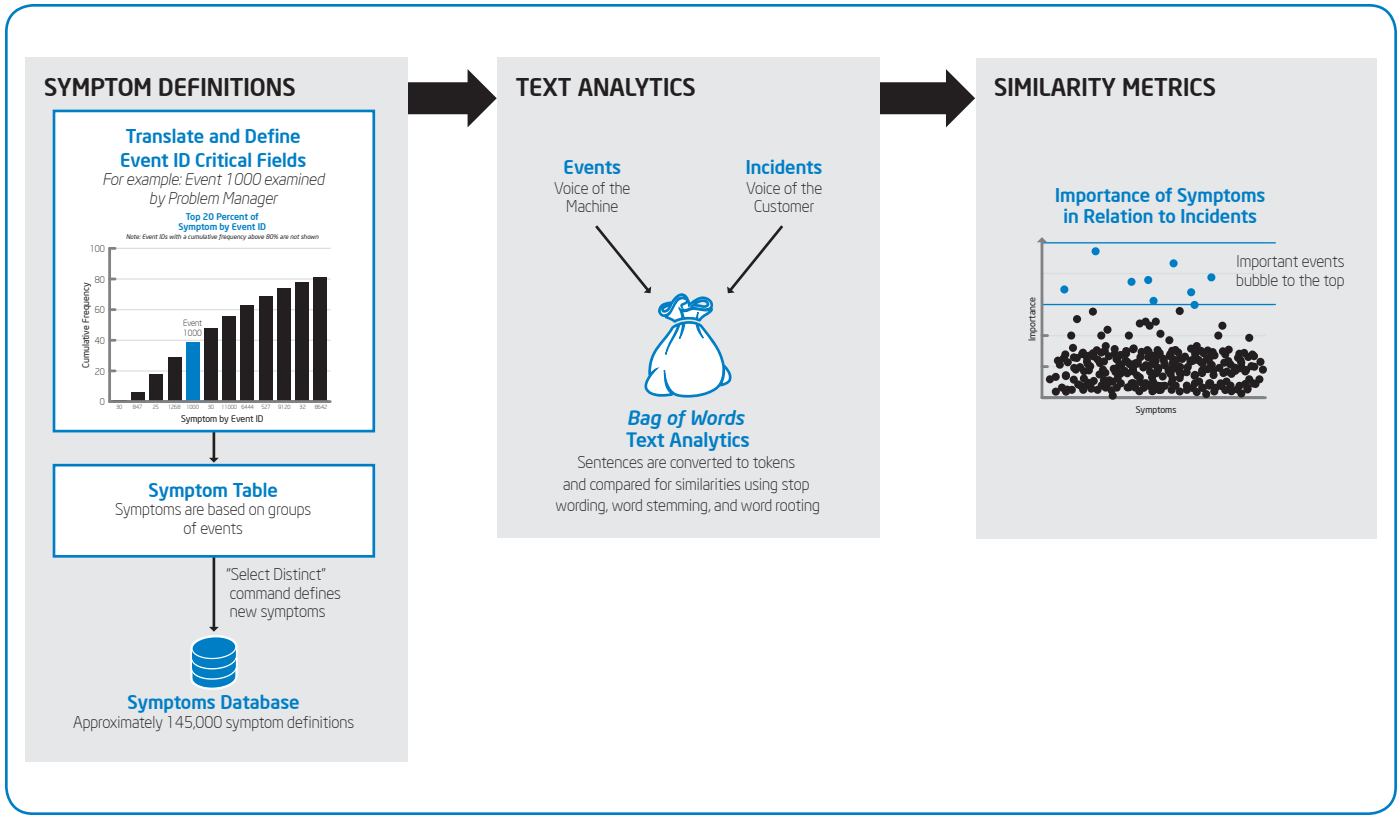

Figure 4. One part of our solution consists of grouping events with similar characteristics into "symptoms." By combining many events into a single symptom, we create a more manageable data set. In the text analytics stage, we compute the similarity between incidents and events, which we can then use to determine the importance of symptoms by the number of associated incidents.

Figure 5 is a heatmap view of all the symptoms by application, including detailed information of each symptom and the associated incidents. The larger the box a symptom commands on the heatmap, the more incidents associated with it.

Data is classified by symptom ID, system manufacturer, client name, operating system version, geographic location, and other key information. This makes it easy to investigate a relationship, such as whether a symptom is associated with a specific computer model (see Figure 6).

A problem manager can filter the data to see which machines with a particular processor are experiencing a particular symptom. This was a difficult association to make when incidents and event data were independent, but now, through our incident predictability model, it is easy to visualize such associations, draw conclusions, and act on them.

Drilling down further reveals the details of the incident. The similarity for this incident with an event, based on the Jaccard similarity coefficient, is very high (0.80). This means that the incident and the event used common words.

RESULTS

Through the application of big data solutions, NLP, and statistical modeling, our client-incident-predictability PoC demonstrated the ability to predict 20 percent of the incidents that appeared in the following 28 days. Our ability to sort through millions of events and thousands of incidents to achieve 78-percent accuracy in predicting future incidents is expected to provide significant cost avoidance to Intel.

The move to proactive client problem management will reduce the current baseline of incidents per week that our problem managers face by approximately 20 percent per week. The gains in employee productivity through fewer incidents, reduction of lost work from system crashes, and greater employee confidence in their clients contribute additional value.

As for incident prediction accuracy, Figure 7 provides a snapshot of data taken in one four-week period that was then used to predict incidents that would appear in the next four weeks. The PoC predicted that 150 incidents would occur. The actual total of incidents was 220. Statistically, this 78-percent accuracy (computed as the R-square of the regression model) is considered very close for a prediction.

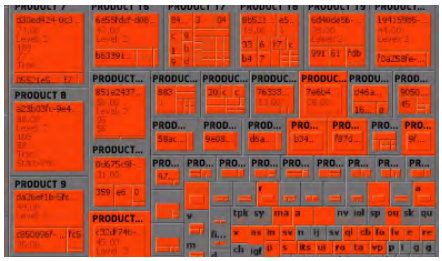

Figure 5. Heatmap views of symptoms by application include detailed information of each symptom and associated incidents. The larger the box a symptom commands on the heatmap, the more incidents associated with it.

Figure 6. Filtering by notebook model brings up a diagram where colors represent models associated with a particular symptom. Drilling down to incident details reveals the degree of similarity between an incident and an event.

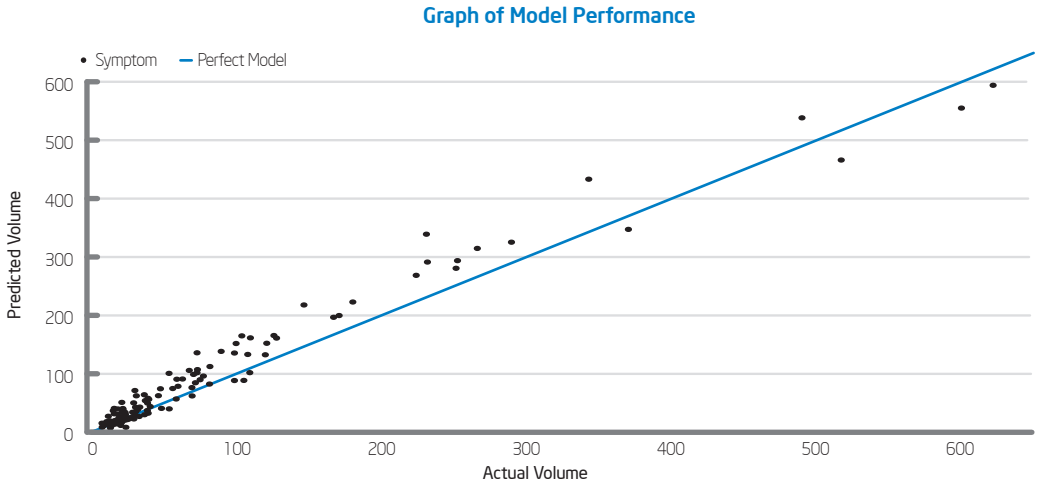

Figure 7. A snapshot of the data taken in the three-week period following the proof of concept shows the accuracy. All of the symptoms with associated incidents (each point in the graph) show predictions that are close to the actual volume of incidents that were actually associated with each symptom.

If each week problem managers implemented solutions for the incidents predicted by our system, they would significantly reduce the number of issues reported to the help desk.

It should be noted that the most time-consuming phase in our PoC was data preparation. Approximately 60 to 80 percent of the project effort centered on understanding the data and then addressing data cleansing, data loading, and data movement. Once those tasks have been accomplished, the rest of the solution required little administrative time.

NEXT STEPS

Our PoC demonstrates our ability to predict with a high degree of accuracy a substantial proportion of the incidents that are likely to occur in the following month.

To further improve our ability to proactively manage the predicted incidents, keeping them from actually turning into incidents, we are taking the following steps:

- Implementing more techniques from advanced NLP that may improve the reliability of our text analytics model that computes the similarity between incident and events
- Developing business processes for proactive problem management based on predicted incidents
- Quantifying incident reductions resulting from our predictive analytics effort to assess improvement and value
- Developing concurrent statistical models and algorithms to reduce false positives (an incident and event that the model erroneously states are associated) and false negatives (an incident and event that should be associated, but the model fails to associate them) and progress to prescriptive analytics

CONCLUSION

Client incident prediction for problem management brings predictive analytics to the service desk, enabling proactive rather than reactive response to client problems. For Intel IT, the ability to predict client incidents potentially means a 20 percent reduction in the current baseline. This could mean approximately 20 percent fewer incidents per year and substantially contribute toward helping us reach our goal for significant incident reduction on all devices across the enterprise.

A common goal in IT service management—and Intel IT is no exception—is to see the same incident no more than two times. Once is an accident, twice is a pattern. Through the use of new big data technologies, advanced text analytics, predictive analytics, and data visualization tools, Intel IT is enabling problem managers to anticipate client problems, find the causes, and address them before they become widespread.

By combining data mining and predictive analytics, our client-incident-prediction solution allows us to derive value from what was once largely ignored data. This data enables us to solve many client issues before they have an impact on user productivity. The ways we use Intel Distribution, Java, Hive, and NoSQL tools with predictive analytics algorithms will likely prove promising in finding new value in data logs for many other data types. We anticipate applications for all or parts of this solution in Intel's manufacturing, supply chain, marketing, market research, and other operations.

RELATED INFORMATION

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

- "Improving Client Stability with Proactive Problem Management"
- "Intel IT Best Practices for Implementing Apache Hadoop* Software"

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

ACRONYMS

JAR	Java archive
NLP	natural language processing
PoC	proof of concept
RDBMS	relational database management system
VOC	voice of the customer
VOM	voice of the machine