Developing a big data application for data exploration and discovery

Tips, techniques, and practical guidelines to help you get started

Seeling Cheung  
Luciano Resende  
Scott Lindner  
Cynthia M. Saracco  
April 23, 2013

Exploring big data and traditional enterprise data is a common requirement of many organizations. In this article, we outline an approach and guidelines for indexing big data managed by a Hadoop-based platform for use with a data discovery solution. Specifically, we describe how data stored in IBM's InfoSphere® BigInsights™ (a Hadoop-based platform) can be pushed to InfoSphere Data Explorer, a sophisticated tool that enables business users to explore and combine data from multiple enterprise and external data sources.

Introduction

If you've been following many of the early case studies around big data, you may have come to believe the saying that "you don't know what you don't know." Indeed, big data applications often focus on gleaning business insights from data that might otherwise be discarded or ignored for a variety of reasons. Increasingly, companies are looking to develop a comprehensive information management strategy that involves more than simply exploring or analyzing big data. Specifically, they want to integrate big data into their overall information management strategies alongside existing data systems, including relational DBMSes, enterprise content management systems, data warehouses, etc.

This article examines one facet of that challenge, outlining an architecture and approach for indexing big data and traditional data sources, as well as providing a web-based interface for discovering new insights across these disparate data sources. In particular, it describes how Data Explorer, a data discovery platform, can index data managed by InfoSphere BigInsights, enabling persistent forms of big data to be combined with existing enterprise data. Both Data Explorer and BigInsights are key components of IBM's big data platform, so let's start with an overview of this platform and these two key offerings.
Overview of IBM's big data platform

IBM's big data platform is designed to help organizations explore, analyze, and manage a wide range of data, including streaming data, traditional business data, and "unconventional" or auxiliary data that previously had been difficult to incorporate into business intelligence and analytical platforms for the enterprise. Let's look at this platform briefly before we focus on two key components — InfoSphere Data Explorer and InfoSphere BigInsights — for the remainder of this article.

Figure 1 depicts the architecture of IBM's big data platform, which differs from other commercial offerings in the breadth of its capabilities. Working from the top down, you'll see that IBM's platform features tools and technologies for visualizing and discovering insights across various data sources, developing analytical applications, and managing your environment. Data Explorer provides key visualization and discovery capabilities for IBM's big data platform, so we'll discuss that component in greater detail shortly. The accelerators shown in Figure 1 are IBM-supplied tool kits that include dozens of pre-built software artifacts to help companies quickly deploy solutions for analyzing social media and machine data (i.e., log records). Three data processing engines enable organizations to work effectively with the variety, volume, and velocity inherent in big data. These engines include a Hadoop-based system (BigInsights, which we'll discuss in greater detail later), a stream computing platform (InfoSphere Streams), and a data warehouse platform (such as PureData™ for Analytics or DB2®). Finally, IBM's big data platform includes connectivity to other popular enterprise software, including relational DBMSes, extract/transform/load platforms, business intelligence tools, content management systems, and more.

Figure 1. IBM's big data platform architecture

Overview of InfoSphere BigInsights

InfoSphere BigInsights is IBM's platform for persisting and analyzing many forms of big data. Based on the open source Apache Hadoop project, BigInsights is designed to help companies discover and analyze business insights hidden in large volumes of data that might otherwise be ignored or discarded because it's too impractical or difficult to process using traditional means. Examples of such data include log records, click streams, social media data, news feeds, email, electronic sensor output, and even some transactional data.
To help businesses efficiently derive value from these types of data, BigInsights Enterprise Edition includes several open source projects from the Hadoop ecosystem, as well as a number of IBM-developed technologies that enhance and extend the value of this open source software. As Figure 2 indicates, these technologies range from application accelerators to analytical facilities, development tools, platform improvements, and enterprise software integration. For example, BigInsights customers can use sophisticated text analytics capabilities to extract content and context from documents, email, and messages. Application developers can employ Eclipse-based wizards to speed development of custom Java™ MapReduce, Jaql, Hive, Pig, and text analytics applications. Administrators can manage and monitor their BigInsights environments through an integrated Web console, and business users can launch IBM-supplied or custom-developed applications through a Web-based catalog.

In this article, we'll focus on a subset of BigInsights features, such as text analytics and application life-cycle tools. For more information about BigInsights, see Related topics.

Figure 2. InfoSphere BigInsights architecture

Overview of InfoSphere Data Explorer

InfoSphere Data Explorer allows you to index large sets of structured, unstructured, and semi-structured data from disparate data sources. It also provides the ability to build big data exploration applications and 360-degree information applications. InfoSphere Data Explorer allows user to create a view of relevant information about different entities like customers, products, events, partners, etc. from large sets of data stored in different internal and external data repositories without having to move data.

A key challenge in today's enterprises is that users can't quickly find the information they need to solve a business problem or complete a task. Often, data is scattered throughout different systems to support specific applications managed by different organizations. In addition, new data sources are emerging as critical resources that people may need to consider in their day-to-day work and make important decisions, such as social media, feeds from mobile devices, Twitter, etc.
As an example, customer information such as contact information, products purchased, service tickets opened, and warranty information are stored in different business applications like CRM, support ticketing systems, marketing portal, etc. Imagine a salesperson who wants to call a customer for an up-sell. He has to first log in to 10 applications to aggregate the data about the customer or talk to five people to understand all these pieces of information.

Data Explorer addresses this key challenge. Information is stored in many different systems and silos, yet users need a consistent way to view all data and quickly navigate to what is most relevant to them. The challenge is delivering information at the point of impact where employees need it most to make decisions.

Figure 3. InfoSphere Data Explorer architecture

Integration of BigInsights and Data Explorer

BigInsights and Data Explorer complement one another, enabling organizations to broaden the scope of information they can analyze in a consistent, coherent manner. For example, BigInsights is often used to store unstructured and semi-structured content. Furthermore, the need to explore and navigate the content is becoming more critical — often in a search-like interface. This allows the information to be more consumable to the line-of-business user. For example, if you are storing machine data, an end user may want to navigate dates of content, look for specific machine failure types, etc. On the other hand, if storing social data, an end user may want to search on user sentiments related to products. All of this requires a rich indexing capability. In addition to indexing, Data Explorer can provide a rich user experience, incorporating content from BigInsights and other enterprise content to allow for full big data exploration.
Sample scenario

To implement such architecture, we need to take several steps. We'll summarize these here, and explore them in greater detail later:

- Collect and prepare your social media data for analysis
  - BigInsights provides a variety of data collection mechanisms through pre-built applications. Once the text-based social media posts reside in BigInsights, you need to extract information of interest so it can be easily indexed and explored later. BigInsights provides sophisticated text-analytics capabilities to help you extract your entities of interest including product, people and sentiments on products.
- Model business entities and relationships of interest
  - An application can jumpstart this process by specifying an entity model for Data Explorer to help set up various configuration options that we will show shortly. This entity model is critical to the overall success of your application scenario.
  - The entity model will capture the set of important business entities and relationships that your business analysts will be interested for searching, discovery and exploration in Data Explorer. Thus, an effective entity model design will presume an understanding of how and what the business analysts would like to search and explore.
  - The entity model will capture the set of important configurations of your Data Explorer cluster to reflect your capacity and deployment planning. A little later, you'll see how we capture products and tweets as key business entities of interest, further specify the relationships among these entities, and provide the topology deployment information of the Data Explorer cluster.
- Develop your first indexing application to index extracted social data into Data Explorer
  - You will be ready to develop your indexing application by leveraging the BigInsights application development life cycle, which allows you to create, publish, and deploy your application with minimal effort. Once deployed, the entity extraction from your social data will be pushed to a Data Explorer search collection, and will be ready for further exploration using the Data Explorer faceted search feature and for building a 360-degree view application.
- Using Data Explorer for visualization
  - Data Explorer Application Builder provides a way to build an application that brings together the relevant information about data spread out among different systems. In our sample scenario, a product planning executive may care about a product or family of products, so a 360-degree view application may include feedbacks, product issues, and past customer interactions.

Collect and prepare your social media data in BigInsights for analysis

BigInsights provides a variety of data collection mechanisms through pre-built applications, such as the Boardreader applications.
You can collect your social data and store them in BigInsights, leveraging a number of storage choices, including the distributed file systems and storage engine like HBase.

Once the text-based social media posts reside in BigInsights, you need to extract information of interest so it can be easily indexed and explored later. BigInsights provides sophisticated text analytics capabilities to help you extract sentiments related to products and extract social media user profiles. The following figure shows snippets of output of extraction of entity and sentiment from the social data, highlighting some fields of interest for this entity extraction, including Category, Brand, Product, Source, IsSentiment, IsCustomerOf, Polarity, Created Time, FullName, Screenname, UserID, Text.
Design and manage your application entity model

Once you have the entity extraction from BigInsights text analytics from the prior section, you will be ready to design the Data Explorer entity model.

This section covers a set of elements in your entity model to consider in your solution and design them into your model. This process will ensure that your application solution will meet the data access and exploration pattern your business analysts need to deliver a scalable search environment for your big data. We will summarize the set of steps in this design process and explore them in more detail shortly:

• Determine the important set of business entities and relationships you are interested to support for further search and exploration in Data Explorer and identify the variety of sources that will be providing that these business entities may be spread out in.
• Capture these entities and relationships in the entity model of your scenario.
• Determine the scalability of your Data Explorer cluster and design these specifications into your entity model. These specifications will determine the scalability of your Data Explorer deployment.
• Deploy your entity model to the Zookeeper cluster for centralized management of configuration settings.

Determining the set of important business entities and relationships to build contextual information

In our sample scenario, we have accumulated internal data about existing customers and our products. This data is stored in a relational DBMS. In addition, there are social tweets we collected and that we have extracted the user sentiments of our products using BigInsights text analytics. Our business analysts may want to obtain a more comprehensive view of how customers perceive our products and how visible our products are in the overall marketplace. Combining enterprise and social media data can thus provide greater insights for our business analysis. We identify that the following entities will be of great interest for our business analyst:

• User sentiment of products extracted from social data in BigInsights
• Product data stored in a relational database
• Online customers stored in a relational database

Equally important is the need to provide the right contextual information for business analysts. To achieve this goal, you need to define the set of relationships among the entities. Relationship is the crucial element Data Explorer Application Builder uses to link the interaction among entities and provides the critical benefit of building the contextual information. For example, in our scenario, we need to capture the fact that tweets are associated with specific users (customers), and some tweets may relate to products.
Capturing these entities and any important relationship in the entity model

The Data Explorer entity model is in an XML format. Use an XML editor of your choice to create a new file for the entity model:

- Adding entity `sentiment`:
  - The snippet to add the sentiment entity to the entity model will look like the following listing. It includes the additional information for the fields we want to capture to allow them to be used in the search application that Data Explorer Application Builder can build.

  **Listing 1. Entity `sentiment` definition**
  ```xml
  <entity-definition default-searchable="true" identifier="@hash"
  name="tweet" store-name="tweet-search-store">
    <field external-name="Category" name="Category"/>
    <field external-name="Brand" name="Brand"/>
    <field external-name="IsSentiment" name="IsSentiment"/>
    <field external-name="Polarity" name="Polarity"/>
    <field external-name="CreatedTime" name="CreatedTime"/>
    <field external-name="Screenname" name="Screenname"/>
  </entity-definition>
  ```

- Adding entity `products`:
  - The snippet to add the product entity and related fields to the entity model will look like the following listing:

  **Listing 2. Entity `product` definition**
  ```xml
  <entity-definition default-searchable="true" identifier="@hash"
  name="product" store-name="product">
    <field external-name="BRAND" name="BRAND"/>
    <field external-name="PRODUCT_NUMBER" name="PRODUCT_NUMBER"/>
    <field external-name="PRODUCT_BRAND_CODE" name="PRODUCT_BRAND_CODE"/>
    <field external-name="PRODUCT_DESCRIPTION" name="PRODUCT_DESCRIPTION"/>
  </entity-definition>
  ```

- Adding the essential relationships among our entities:
  - Relationship is the crucial element that Data Explorer Application Builder uses to link the interaction between entities and provides the critical benefit of building the contextual information. As an example for our scenario, we may want to capture the fact that some tweets are related to products. The relationship definitions may look like the following listing:

  **Listing 3. Relationship definition**
  ```xml
  <entity-definition default-searchable="true" identifier="@hash"
  name="product" store-name="product">
    ...
    <association-definition name="feedback" to="FEEDBACK_TYPE">
      <link from-field="BRAND" fuzzy="false" to-field="Brand"/>
    </association-definition>
  </entity-definition>
  ```

Provide topology specifications for your Data Explorer cluster

- Specifying cluster collection store for Data Explorer:
Once you've identified entities of interest and relationships, you need to build an index to support search, discovery, and analysis. To do so, you need to specify a storage mechanism — a collection store — for this index. For use cases involving BigInsights, such as our scenario, we would want to use a cluster collection store. It's one of several types of collection stores supported by Data Explorer. Choosing the cluster collection store type will enable the Data Explorer engine to leverage a cluster of machines to scale horizontally to handle larger scale of indexing for BigInsights data.

The following snippet shows how to specify the cluster collection store for indexing social data coming from BigInsights. The other entity representing the data from relational DBMS will use the more typical single collection.

**Listing 4. Cluster collection store for BigInsights data**

```xml
<cluster-collection-store activity-collection="false"
    collection-name="tweet-search-store"
    monitor-activities="false"
    name="tweet-search-store"
    base-collection="default-push"
    n-shards="2"/>
<collection-store activity-collection="false"
    collection-name="gssdb-product"
    monitor-activities="false"
    name="product"/>
```

Specifying scalability of your Data Explorer cluster:

- Adding shards to your search application allows data to be partitioned horizontally, particularly when these shards are spread across multiple physical Data Explorer instances. Overall performance can be increased when handling large amounts of data as indexing and search operations are distributed into a clustered environment. Reusing the entity model example from above, we specify the number of shards in our cluster collection store and spread them between two different physical Data Explorer instances.

**Listing 5. Specifying scalability for BigInsights data**

```xml
<velocity-instance url="http://velocity1.domain.com:9080/vivisimo/cgibin/velocity?v.app=api-soap&amp;wsdl=1&amp;use-types=true&amp;"
    username="api-user"
    password="password">
    <serves name="tweet-search-store"
        shard="1"
        n-shards="2"
        port="9081"/>
    <serves name="tweet-search-store"
        shard="2"
        n-shards="2"
        port="9082"/>
</velocity-instance>
<velocity-instance url="http://velocity2.domain.com:9080/vivisimo/cgibin/velocity?v.app=api-soap&amp;wsdl=1&amp;use-types=true&amp;"
    username="api-user"
    password="password">
    <serves name="tweet-search-store"
        shard="1"
        n-shards="2"
        port="9081"/>
    <serves name="tweet-search-store"
        shard="2"
        n-shards="2"
```

Developing a big data application for data exploration and discovery

Page 9 of 20
Using ZooKeeper to manage your Data Explorer entity model

Data Explorer uses ZooKeeper to manage the entity model of your application. Zookeeper is a central service for maintaining configuration information, providing distributed synchronization, and providing group services. Now that we have defined our application entity model, we need to make it available to the application by uploading to a ZooKeeper cluster. This configuration cluster will be used by the application to discover the deployment topology in use:

- Uploading your entity model to ZooKeeper cluster:
  - Once your ZooKeeper cluster is set up, you can upload to and manage your Data Explorer application entity model using this ZooKeeper cluster. The bigindex JAR included in the lib folder of Data Explorer BigIndex API ZIP is an executable that can be used as a basic command-line tool to upload and manage the entity model in ZooKeeper. The usage of the command line is shown below.

Listing 6. Uploading the entity model to the ZooKeeper cluster

```java
java -jar bigindex-2.0.0.jar
  --properties-file zookeeper.properties
  --import-file scenario_entity_model.xml
  --export-to-screen --legacy-model
```

Note that if you use the Data Explorer Application Builder administrative UI to manage your application entity model, you can skip the step above and alternatively point your application to the same ZooKeeper server instance and namespace used by Data Explorer Application Builder. You can find more details about the ZooKeeper configuration being used at the zookeeper.yml located in IBM/IDE/AppBuilder/wlp/usr/servers/AppBuilder/apps/AppBuilder/WEB-INF/config.

Developing your first BigInsights indexing application with Data Explorer

Once you are done with designing your Data Explorer entity model, you are ready to leverage the BigInsights application development life cycle to develop your first indexing application to push your social data into a search collection in Data Explorer. The BigInsights application framework allows you to create, publish, and deploy your first indexing application with minimal effort.

Create a BigInsights project and create a new Java class

You need to create an appropriate project for your application, as you might expect from any Eclipse-based application development effort, check out the article "Developing, publishing, and deploying your first Big Data application with InfoSphere BigInsights" for the quick steps to create a BigInsight project (see Related topics). After you create the BigInsights project, you will need to add a new Java class to the project. To do so, from your Eclipse environment, select File > New > Java > Class. Fill in the information for your class (package name, etc.) and when you are done, click Finish.
Developing a big data application for data exploration and discovery

Using new BigIndex APIs to index BigInsight data into Data Explorer

Your application will invoke a set of indexing Java APIs provided by Data Explorer (BigIndex APIs) to push the data from BigInsights. The following steps show the various key pieces of the APIs to accomplish this goal:

• Retrieving the Data Explorer deployment topology for indexing:
  • As you recall from the prior section on entity model, the Data Explorer cluster topology is captured in the entity model uploaded to ZooKeeper. Your indexing application will need to establish a connection to this ZooKeeper cluster to retrieve the topology and locate the Data Explorer cluster for indexing. The following listing is a code snippet that achieves this task.

Listing 7. Establishing connection to ZooKeeper cluster

```
ZookeeperConfiguration zookeeperConfiguration =
    new ZookeeperConfiguration("namespace_sample_big_data_app",
    new ZookeeperEndpoint("zkhost1.domain.com", 2181));
```

• Using a field parser to process your input data format:
  • Once the Data Explorer instance we will use for indexing is established, we are ready to process our input data for indexing. For our sample scenario, the social data is captured in CSV format. We may need to use the open source OpenCSV parser to parse each CSV file and process each row into a list of key values. This will prepare the data into the right format for the Data Explorer indexing engine. Below is a code snippet to give a sample of the application logic for parsing CSV data. Note: You will need to use the appropriate field parser to process the data format of your input data for indexing.

Listing 8. Sample code for parsing CSV data

```
// Read each CSV input file stored on BigInsights HDFS
for (FileStatus fStatus : listFilesFromHDFS(inputDirectory)) {

  // For each CSV file, parse each row into a list of key values
  CSVReader reader =
    new CSVReader(new InputStreamReader(fs.open(fStatus.getPath())));

  // For each key, we will show later how to index it into
  // the Data Explorer index record
  while((listOfFields = reader.readNext())!= null){
    ...
```

• Defining your record schema for sample social data:
  • The Data Explorer indexer also expects the application to define the schema of an index record. The following listing shows the Java code snippet using the Data Explorer BigIndex API to define the record schema for the various key fields of our Tweet data. Note: In the following call to addRecordType(), the input value must match the name of the entity as defined in your entity model. In our scenario, the entity name is "tweet."
Listing 9. Defining record schema

```java
RecordSchema recordSchema = new RecordSchemaBuilder()
    .addRecordType("tweet")
    .addTextField("Category").retrievable(true).sortable(true)
    .addTextField("Brand").retrievable(true).sortable(true)
    .addTextField("Product").retrievable(true).sortable(true)
    .addTextField("isSentiment").retrievable(true).sortable(true)
    .addDateField("CreatedTime").retrievable(true).sortable(true)
    .addTextField("Screenname").retrievable(true).sortable(true)
    .build();
```

• Indexing the records into the Data Explorer engine:
  • Now that you are done defining the schema of a record, you are ready to add each field of the record to the index and continue to the next record until you are done indexing all the records in your social tweet data. Below is the code snippet to show indexing the records into the Data Explorer engine. **Note:** In the call to `newRecordBuilder()`, the value must match the name of the entity as defined in your entity model. For example, in our sample scenario, the entity name is “tweet.”

Listing 10. Indexing records

```java
RecordBuilderFactory recordBuilderFactory = new RecordBuilderFactory(recordSchema);

// In the following call, provide the name of
// the entity as defined in the entity model
RecordBuilder recordBuilder = recordBuilderFactory.newRecordBuilder("tweet");

// For each CSV row that's been parsed into a list of fields
while((listOfFields = csvreader.readNext())!= null){
    recordBuilder.id(String.valueOf(recordId++));
    // For each field, set field name and field value
    while (i != listOfFields.length){
        String fieldName = listOfFieldNames[i];
        String fieldValue = listOFields[i];
        // Add the field to the indexing record
        recordBuilder = recordBuilder.addField(fieldName, fieldValue);
    }
}

// Finally, call to generate the record with the
// current data and add it to the indexer
RequestStatus status = indexer.addOrUpdateRecord(recordBuilder.build());
```

Publish and deploy your indexing application

After developing your indexing application, you are ready to publish it in the BigInsights application catalog. Packing and publishing your indexing application enables you to define the application’s workflow, specify parameters like input data (your social media data), and your Data Explorer ZooKeeper endpoint. Check out the article “Developing, publishing, and deploying your first Big Data application with InfoSphere BigInsights" for the quick steps to create a BigInsight project (see Related topics) for an overview of the steps in publishing your BigInsights application. During this publishing process, you would specify the following information for your indexing application:
• Application type:
  • Select the application type workflow, as shown in the following Figure.

**Figure 7. Application type**

```markdown
<table>
<thead>
<tr>
<th>Application Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
</tr>
<tr>
<td>BigSheets</td>
</tr>
<tr>
<td>Jaql Module</td>
</tr>
<tr>
<td>Text Analytics</td>
</tr>
</tbody>
</table>
```

• Oozie workflow definition:
  • The BigInsights web console generates an Oozie workflow to help manage MapReduce jobs. In the workflow tab, accept the default of allowing the wizard to create a new action: workflow.xml. In the drop-down menu, change the workflow type to Java, as shown below.

**Figure 8. Oozie workflow action type**

```
Location Application Type Workflow Text Analytics BigSheets Jaql Module Parameters Publish
```

• Indexing application parameters:
  • On the Parameters page, specify the parameter for your indexing application, including the input directory. Optionally, you can also provide the ZooKeeper endpoint information as an input parameter to the indexing application, instead of having it hard-coded in the application. The final workflow may look like the following figure.
Setting up Data Explorer client libraries in BigInsights cluster

Before running your indexing application, you need to set up certain Data Explorer client libraries in the BigInsights cluster.

- Copy the install-dir/AppBuilder/bigindex.zip folder from the installation of your Data Explorer cluster to the local file system of the BigInsights cluster.
- Uncompress the bigindex.zip folder. You see the list of the Data Explorer dependency JAR files.
- Make a HDFS directory like /biginsights/oozie/sharedLibraries/DataExplorer.
- Copy the Data Explorer dependency JAR files to the HDFS directory /biginsights/oozie/sharedLibraries/DataExplorer using the Hadoop copy command (e.g., hadoop fs - copyFromLocal *jar /biginsights/oozie/sharedLibraries/DataExplorer/), or use the BigInsights Console to upload the files to the HDFS directory.

Monitoring your indexing application

Once your application is deployed, it will appear in the BigInsights web console as shown in the following figure. You can use the web console to inspect details of the application and run it. You can learn more about the BigInsights web console for monitoring your workflow by checking out the article "Exploring your InfoSphere BigInsights cluster and sample applications" (see Related topics).

Figure 10. BigInsights indexing application
Visualizing with Data Explorer

Verifying your social data in the Data Explorer index

Once your social data is pushed from BigInsights to a search collections in Data Explorer, you should be able to use the Data Explorer Engine administrative UI to inspect the indexed data. For example, you can visually verify that the various fields of interest have been indexed accordingly. To access the administrator UI, follow the steps below:

- Log in to Data Explorer Engine administrative UI.
- Select **Search Collection** from the left menu.
- Look for the collection store for the social data as specified in your entity model.
- Open the search collection, and click the **Search** button on the left panel.

**Figure 11. Data Explorer Engine administrative UI**

- Searching for user tweets on product:
  - In the search box, a user can then type in keywords such as `golf` to perform text search using the existing interface as shown in the following figure.
Figure 12. Text search on user tweets related to golf

Leveraging Data Explorer Application Builder

New in Data Explorer is the Application Builder, which provides the framework to build compelling data exploration application such as faceted search as well as 360-degree information applications that can bring together the relevant information about data spread out among the systems.

- Faceted search:
  - The following figure illustrates the search widget that can be built using Application Builder to provide an intuitive faceted search application for users for exploring their social data. Faceted search allows you to easily navigate your result sets on a specific topic using a set of refinements. In this example, we explore users tweets on products such as golf as shown.
• 360-degree information application:
• After you explore aspects of the social data, you can also relate it to additional types of data, such as customer or product data, that can be extracted from other systems. Data Explorer provides connectivity and crawling capability to various relational databases, enterprise CRM systems, file shares, and others. Data Explorer Application Builder provides the way to build a 360-degree view application that brings together the relevant information about data spread out among these disparate systems, all while leaving the data where it originally resides.
In our sample scenario, a marketing analyst may care about a product or family of products, so a 360-degree view application may include user feedback and product details. The following figure illustrates a product page where multiple widgets are brought together, displaying product information with related user comments. For more information about building entity pages with multiple widgets shown here, see Related topics.
Figure 15. 360-degree view application — Entity page

Note that figures 14 and 15 demonstrate a 360-degree application and involve data from different systems that were not mentioned in this article.

Summary

This article explored a software architecture that enables business analysts to explore data derived from a variety of disparate sources with ease and efficiency. In particular, we examined how InfoSphere Data Explorer can index big social media data managed by InfoSphere BigInsights, as well as structured data managed by more traditional enterprise data sources. Indexing the data allows for efficient access, while the faceted search capabilities of Data Explorer provide an intuitive way for non-programmers to explore this data, analyze relationships, and get insight.

Acknowledgements

The authors would like to thank some of the colleagues who worked on this technology and those who contributed ideas to this article. In alphabetical order: Stephen Brodsky, Jean Lange, Stacy Leidwinger, Alex Tambellini, and Tuong Truong.
Related topics

- Refer to the InfoSphere Data Explorer InfoCenter for documentation about the product. In particular, you may find the Creating Applications Using the Data Explorer Engine API and Data Explorer Application Builder sections useful.
- Read "Understanding InfoSphere BigInsights" to learn more about the BigInsights's architecture and underlying technologies.
- Learn about the BigInsights web console and sample applications by reading "Exploring your InfoSphere BigInsights cluster and sample applications."
- Check out "Developing, publishing, and deploying your first Big Data application with InfoSphere BigInsights" for quick steps to create a BigInsight project and a sample application.
- Check out the Big Data: Frequently Asked Questions for IBM InfoSphere BigInsights video to listen to Cindy Saracco discuss some of the frequently asked questions about IBM's Big Data platform and InfoSphere BigInsights.
- Visit BigData University for links to technical materials, demos, training courses, news items, and more.
- Get a trial version of IBM InfoSphere Streams.

© Copyright IBM Corporation 2013
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)