Information Management

IBM

BigInsights Text Analytics Tutorial

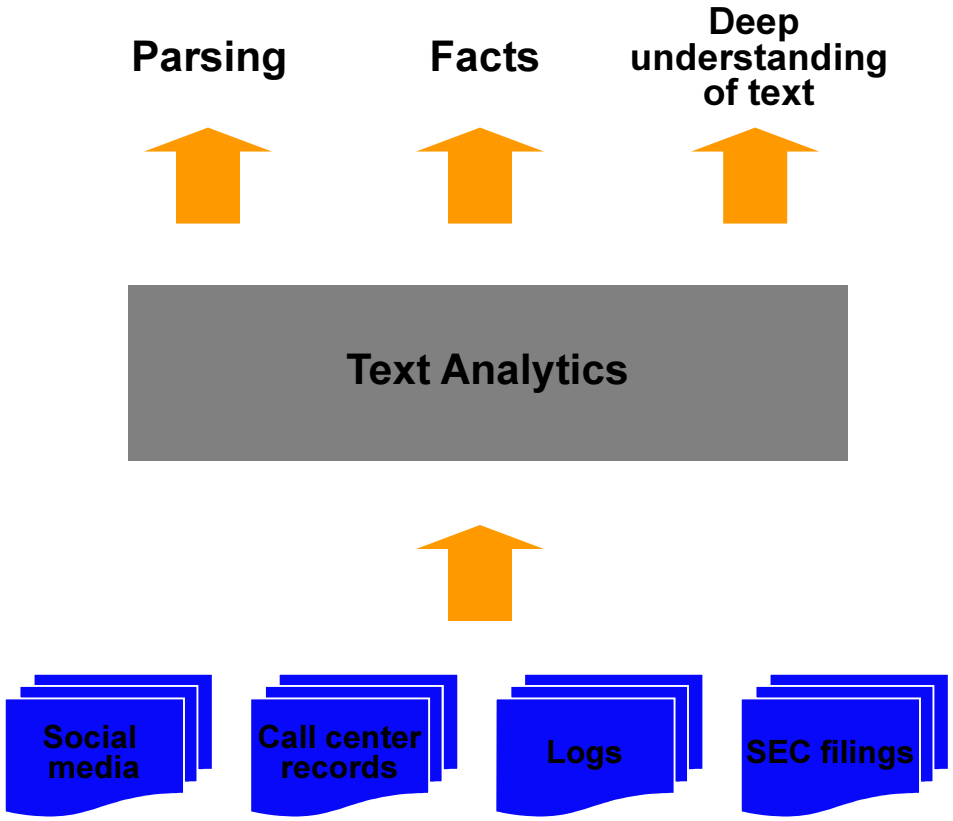
Search & Analytics Group
IBM Research - Almaden

February 8, 2014

Introduction to Text Analytics and BigInsights capabilities

- **Introduction**
  - What is text analytics?
  - Example text analytics applications and data sources
  - Critical success factors
  - Previous approaches
- **BigInsights Text Analytics Overview**
  - Architecture
  - Key advantages
- **Tutorial Overview**

What is Text Analytics ?

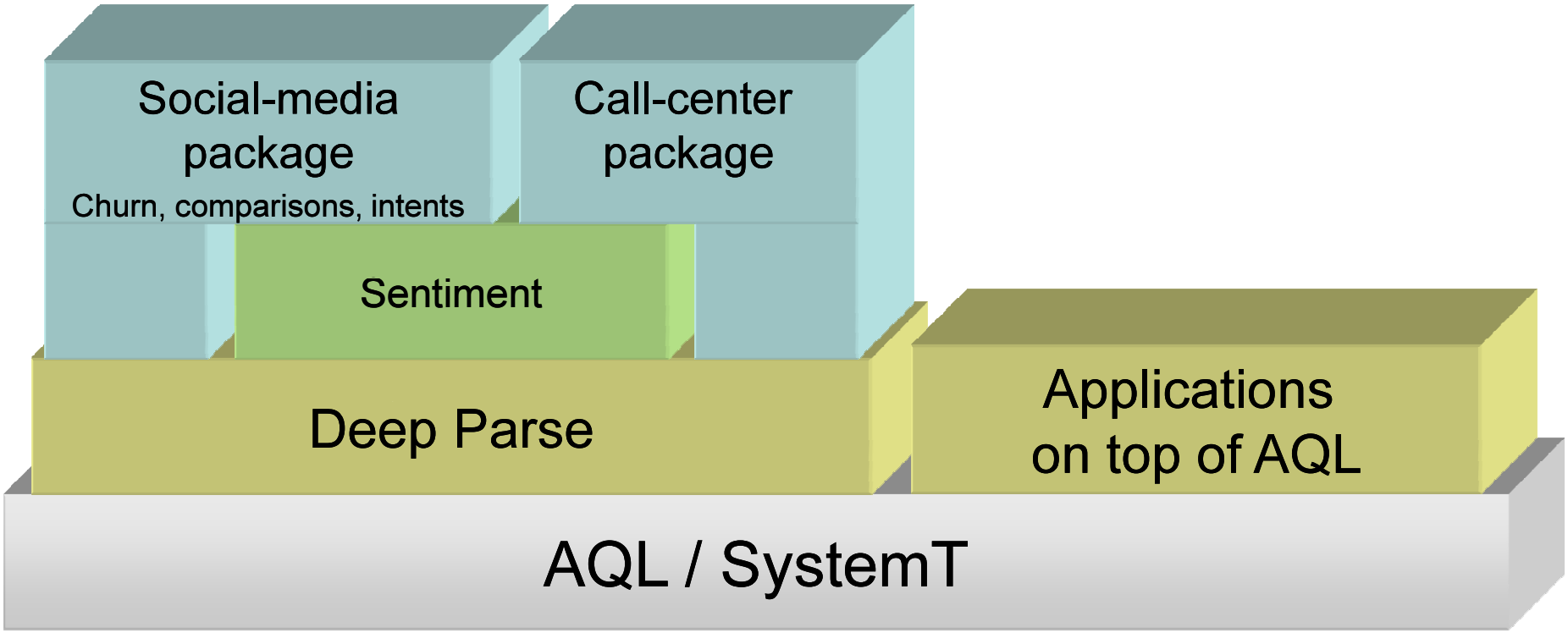


Verticals and Horizontals using Text Analytics

- **Verticals**
  - Financial: financial events, company earnings, key players, etc.
  - Healthcare: drugs / diseases, patient history, chemical compounds
  - General: named entities (person, organization, location, phone, URL, email)

- **Horizontals**
  - CRM Analytics
    - Voice of customer
    - Product / Services gap analysis and in combination with Social Media predicting churn etc.
  - Social Media Analytics
    - Retail applications such as intent identification and customer churn
    - Reputational Risk applications such timeliness of response
  - Digital Piracy
    - Illegal broadcast of streaming and video content
    - Illegal dissemination of copyrighted digital material
  - Log Analytics
    - Log parsing into fields, IP addresses, exception stack trace
  - Data Redaction
    - Identify sensitive information (people names, DOB, SSN)
  - Regulatory Compliance

Building Applications on top of BigInsights Text Analytics: A Conceptual View



Data Sources for Text Analytics are Highly Heterogeneous

- **Variations in content quality: from formal to informal (noisy)**
  - Formal: news reports, financial reports, patent applications
  - Informal: email, blogs, Twitter/Facebook posts

- **Variations in structure: from unstructured to semi-structured**
  - Unstructured: news reports
  - Semi-structured: system/application logs, web pages, financial reports (SEC)

- **Variations in size: from very small to very large**
  - Small (bytes): Twitter posts
  - Medium (Kilobytes): email, blogs, news reports
  - Large (Megabytes): financial reports, patents

Critical Success Factors

- **Quality: Drives effectiveness of entire application**
  - Need high accuracy and coverage
- **Performance: Dominant cost is CPU**
  - Process large documents and large number of documents with high throughput
- **Explainability**
  - Determine the cause of errors and fix it without affecting the remaining correct results
- **Reusability: easily adaptable for a different domain**
  - The development platform must enable layers of abstractions to be built and easily reused in a different domain

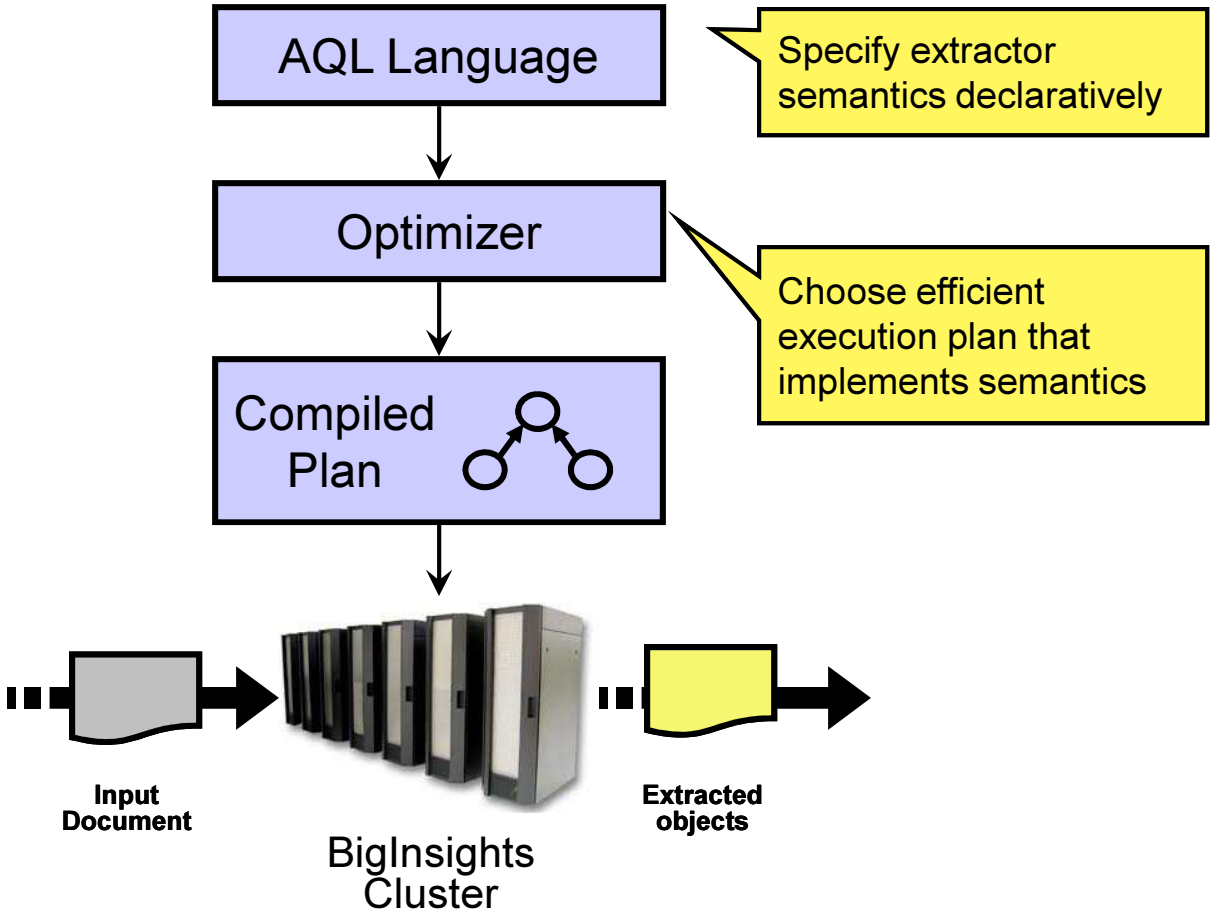
Previous Approaches to Text Analytics

- **Statistical (Machine Learning)**
  - Labeled data required to train a model
  - Model must be retrained for each domain
  - The model is opaque
- **Rule-based**
  - No unified formal language
  - Performance and expressivity limitations

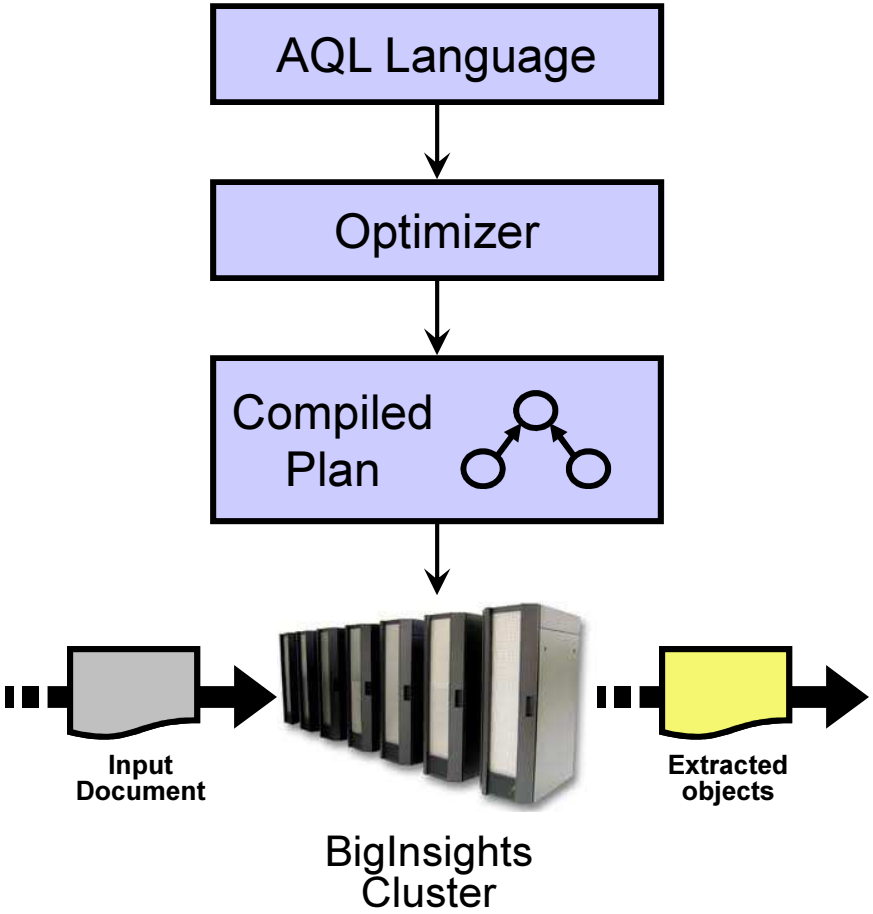
Outline

- **Introduction**
  - What is text analytics?
  - Example text analytics applications and data sources
  - Critical success factors
  - Previous approaches
- BigInsights Text Analytics Overview
  - Architecture
  - Key advantages
- **Tutorial Overview**

BigInsights Text Analytics Architecture



BigInsights Text Analytics Components



- **Eclipse Tools**
  - Develop and maintain extractors in AQL
- **Pre-compiled extractor library**
  - Western languages: Named Entities (person, organization, location, phone, URL, email, date/time) and financial events (merger, acquisition, company earnings)
  - Chinese/Japanese: Named Entities (Person, Organization, Location)
- **Jaql Text Analytics module**
  - Execute extractors on the cluster from Jaql
- **Text Analytics Java API**
  - Invoke Text Analytics directly from your application

AQL: A Declarative Language to Specify Extraction Patterns


```
create view FirstCaps as
select CombineSpans(F.name, C.name) as name
from First F, Caps C
where FollowsTok(F.name, C.name, 0, 0);
```

Choice of SQL-like syntax for AQL motivated by wider adoption of SQL

The Expressivity of AQL

- **Feature Extraction primitives**
  - Regular Expressions
  - Dictionary
- **Text-specific primitives**
  - Span-based predicates
  - Multi-lingual support
    - Tokenization: Arabic, Chinese, Czech, Danish, Dutch, English, German, Greek, Spanish, French, Italian, Japanese, Korean, Finnish, Norwegian (Nynorsk and Bokmal), Polish, Portuguese, Russian, Swedish
    - Parts-of-speech: Chinese, English, German, Japanese, Spanish, French
- **Set-level primitives**
  - Join
  - Block
  - Consolidation
  - Group By
- **AQL Reference in Info Center**
  http://publib.boulder.ibm.com/infocenter/bigins/v1r3/topic/com.ibm.swg.im.infosphere.biginsights.doc/doc/biginsights_aqlref_con_aql-overview.html

Why "Declarative" Language ?

- **Semantics are separate from execution**
  - The developer expresses "**what**" to extract
  - The system determines "**how**"

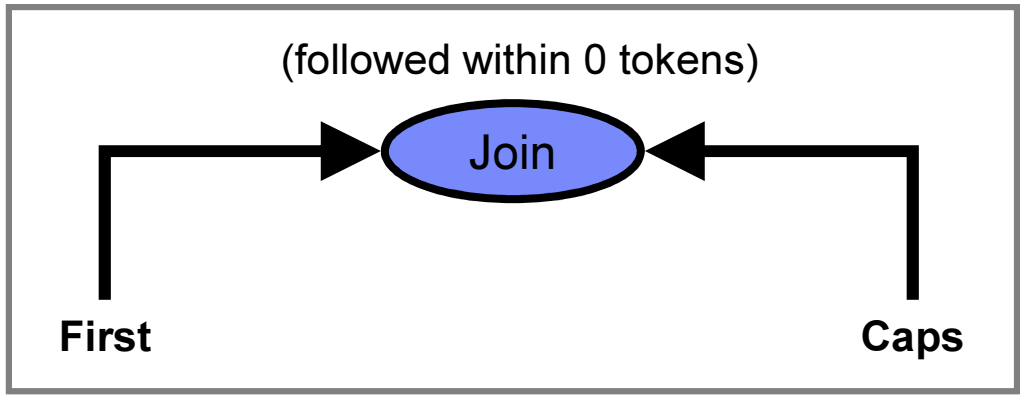
- **Advantages**
  - Explainability
    - Easy to understand and debug
  - Global optimization
    - The system can determine an efficient execution plan
    - The developer does not worry about performance

Scalability of BigInsights Text Analytics

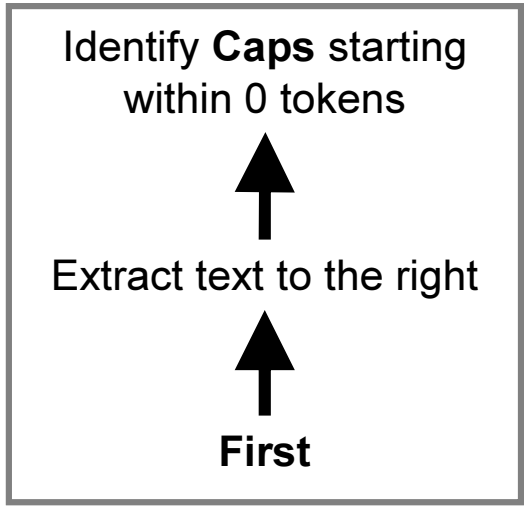
1. Better throughput via query optimization
2. Massive scale-out on BigInsights

Scalability: Flavor of Optimization

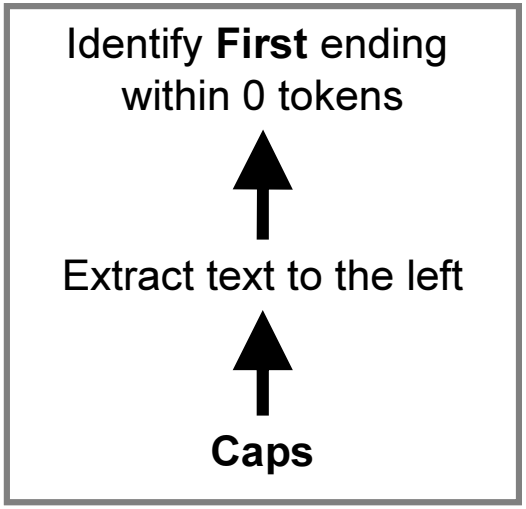
Plan A



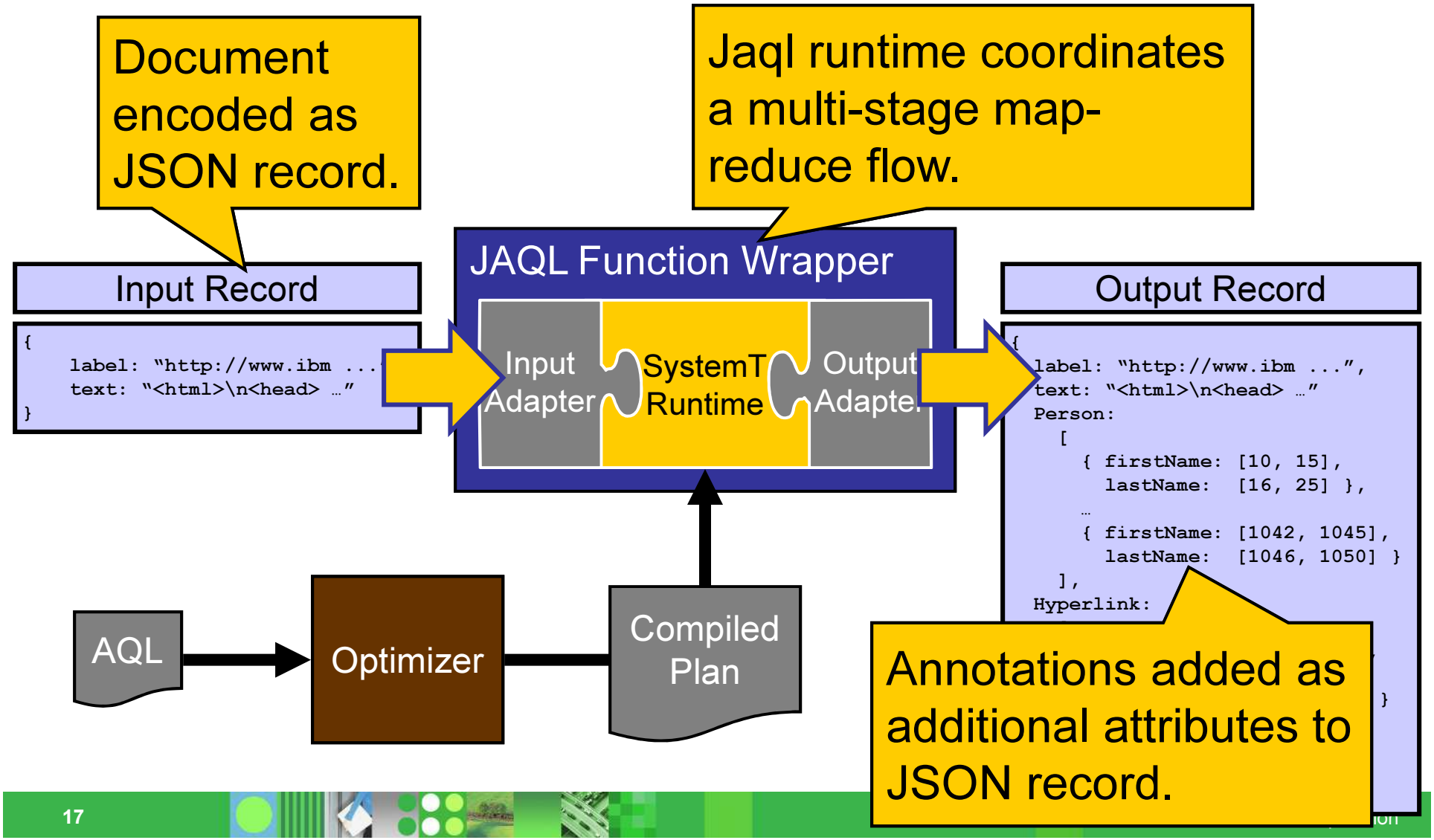
Plan B



Plan C



Scaling Up on BigInsights Clusters



Eclipse Tools Overview

Ease of Programming

AQL Editor: syntax highlighting, auto-complete, hyperlink navigation

Result Viewer: visualize/compare/evaluate

Explain: show how each result was generated

Workflow UI: enable novice users to become experts in a short time

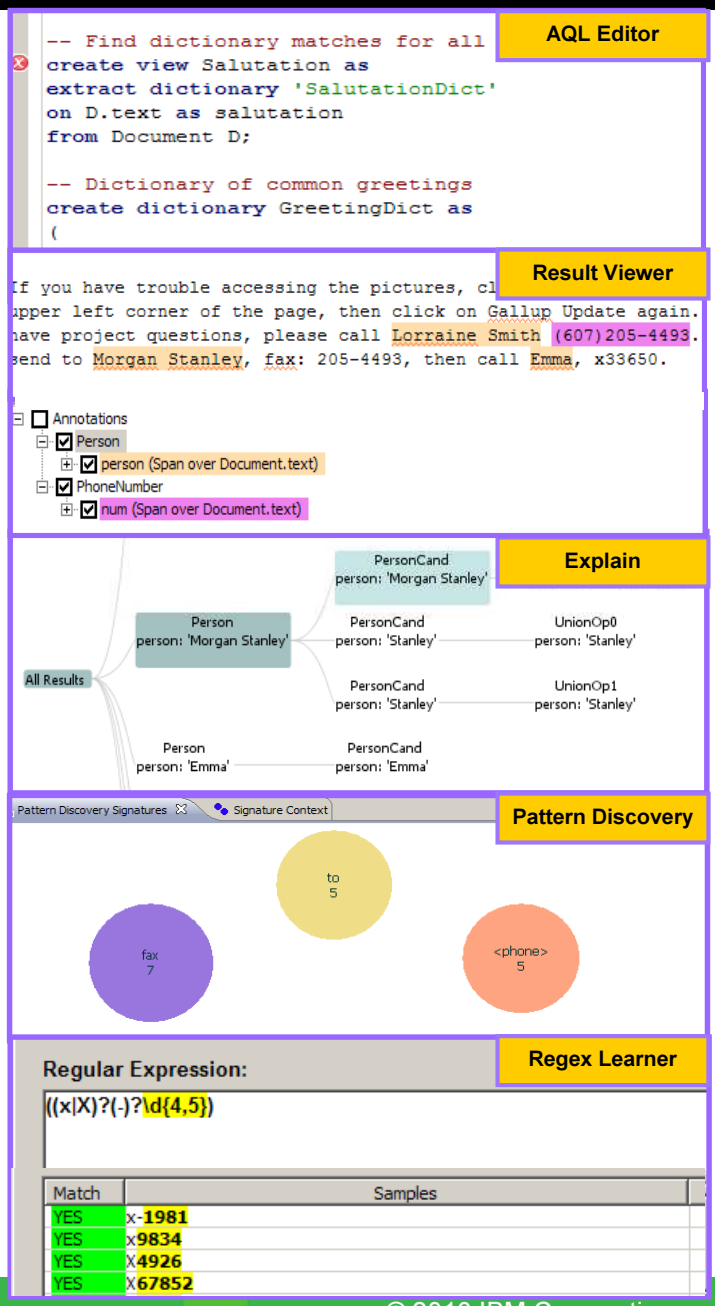
Automatic Discovery

Pattern Discovery: identify patterns in the data

Regex Generator: generate regular expressions from examples

Performance Tuning

Profiler: identify performance bottlenecks to be hand tuned



Key Advantages of BigInsights Text Analytics

- **Quality: Drives effectiveness of entire application**
  - Highly expressive AQL language → Easy to express complex concepts and to improve quality
- **Performance: Dominant cost in text analytics is CPU**
  - Optimizer → Developers don't need to worry about performance
- **Ease of Development, Explainability and Reusability**
  - Looks like SQL → Low learning curve
  - Declarative language → Output can be explained by automatic tools
  - Eclipse Development Tools → Rule editing/discovery

Best of rule-based and statistical approaches!

- Rule-based at Runtime → quality and performance
- Statistical approaches for building rules → ease of development

Outline

- **Introduction**
  - What is text analytics?
  - Example text analytics applications and data sources
  - Critical success factors
  - Previous approaches
- **BigInsights Text Analytics Overview**
  - Architecture
  - Key advantages
- Tutorial Overview

Developing an Extractor with AQL

- **Combinations of syntactic patterns using regular expressions, dictionaries, span operations, relational operators and consolidate**
  - Virtually everything you would need is exposed in the language
  - **Material covered in Days 1 and 2**
- **Different domain → the extractor must be customized**
  - Domain adaptation guidelines in place
  - **Material covered in Day 3**
- **Different language → the extractor must be customized**
  - Language adaptation process worked out with business partner

Existing AQL Extractor Library

- **We developed many extractors by writing AQL rules**
  - Combinations of syntactic patterns using regular expressions, dictionaries, span operations, relational operators and consolidate
  - Examples:
    - Simple entities: numbers, IP addresses, error messages, …
    - Complex entities: person, organization, location
    - Names of drugs, diseases, chemical compounds,…
    - Financial facts: merger/acquisition, earnings, key players,…
- **Adapted some of the extractors for:**
  - Multiple industry verticals: financial, healthcare
  - Multiple data sources: news, email, blogs, log records
  - Multiple languages:
    - Named Entities (person, location, organization): Western languages (DE, EN, ES, FR, IT) , Chinese, Japanese
- **Developed tooling for writing and customizing AQL**
  - Generate regular expressions and discover dictionaries
  - Explain rule output and help fix mistakes
  - Organize complex rule sets
- **Material covered in Days 1-2, and partly Day 3**

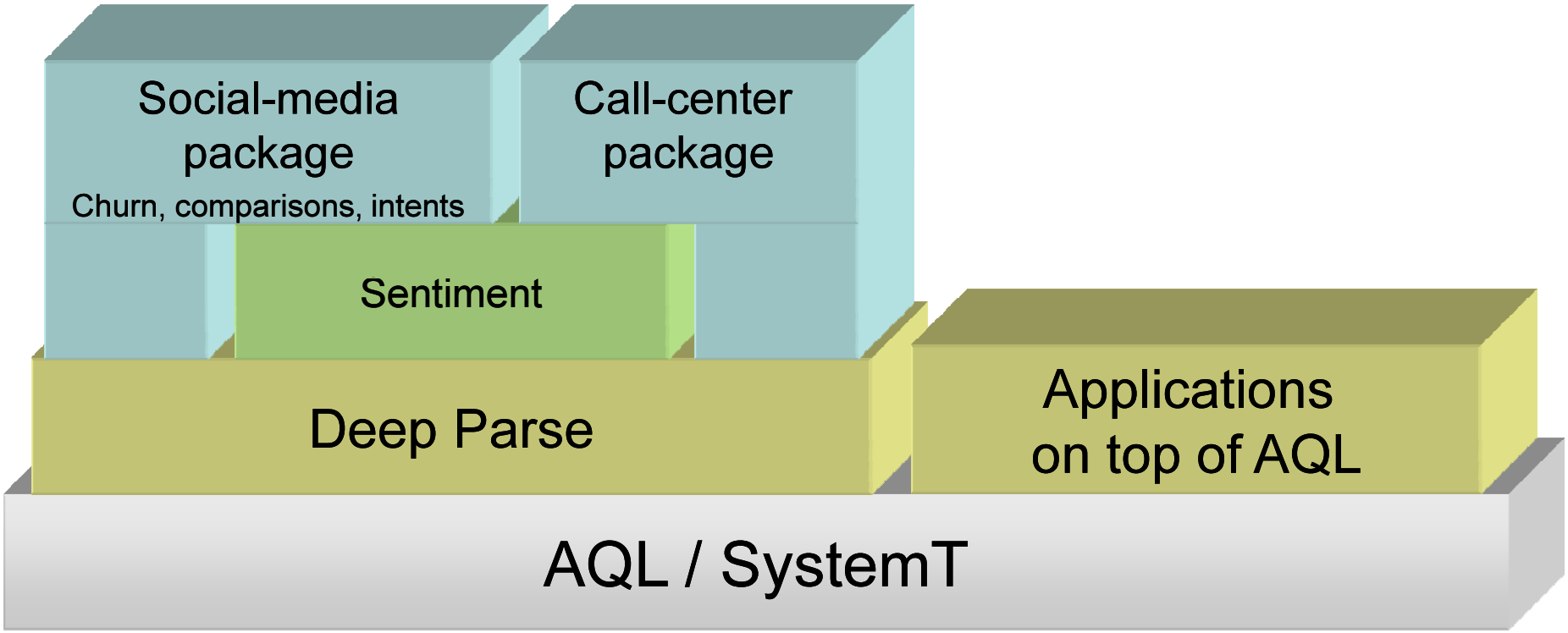
Fact Extraction versus Deeper Understanding

- **A large percentage of use cases can be addressed by using syntactic patterns, building dictionaries, span operations, relational operators and consolidate**
  - Extraction of facts

- **But there is also a need for deeper, semantic, extraction**
  - Involves deeper understanding of linguistics and meaning of text
  - BigInsights Text Analytics is sufficiently powerful for such analysis, but the rules and domain adaptation will be complex, BUT can be learned

- **Need to hide the complexity**
  - We are building higher levels of abstraction for semantic extraction
    - Deep parsing layer built using AQL
    - Sentiment layer built on top of deep parsing layer
  - Working on guidelines for domain adaptation of the two layers
    - **Introduction to this material will be given on Day 3**

Building Applications on top of BigInsights Text Analytics: A Conceptual View



References (1/2) – Peer-reviewed Publications

- **Overview**
  - Rajasekar Krishnamurthy, Sriram Raghavan, and Huaiyu Zhu: "Evolution of Rule-Based Information Extraction: From Grammars to Algebra", Tutorial given at CIKM 2008.
  - Laura Chiticariu, Yunyao Li, Sriram Raghavan, and Frederick Reiss: "Enterprise Information Extraction: Recent Developments and Open Challenges". SIGMOD 2010 (tutorial)

- **Runtime Engine and Extractor Library**
  - Frederick Reiss, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, Shivakumar Vaithyanathan: "An Algebraic Approach to Rule-Based Information Extraction". ICDE 2008: 933-942
  - Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan, Huaiyu Zhu: "SystemT: a system for declarative information extraction". SIGMOD Record 37(4):7-13 (2008)
  - Eirinaios Michelakis, Rajasekar Krishnamurthy, Peter J. Haas, Shivakumar Vaithyanathan: "Uncertainty management in rule-based information extraction systems". SIGMOD Conference 2009: 101-114
  - Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan: "SystemT: An Algebraic Approach to Declarative Information Extraction". ACL 2010.
  - Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, Shivakumar Vaithyanathan: "Domain Adaptation of Rule-based Annotators for Named-Entity Recognition Tasks". EMNLP 2010.
  - Daisy Zhe Wang, Long Wei, Yunyao Li, Frederick Reiss and Shivakumar Vaithyanathan. Selectivity Estimation for Extraction Operators over Text Data. ICDE 2011

- **Tooling**
  - Yunyao Li, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, H. V. Jagadish: "Regular Expression Learning for Information Extraction". EMNLP 2008: 21-30
  - Bin Liu, Laura Chiticariu, Vivian Chu, H.V. Jagadish, Frederick Reiss: "Automatic Rule Refinement for Information Extraction". PVLDB 2010.
  - Yunyao Li, Vivian Chu, Sebastian Blohm, Huaiyu Zhu, Howard Ho. “Facilitating Pattern Discovery for Relation Extraction with Semantic-Signature-based Clustering”. CIKM 2011

References (2/2) – US Patents

- **Runtime Engine**
  - **US Patent Publication 20090198646:** "Systems, Methods and Computer Program Products for an Algebraic approach to Rule-based Information Extraction"
  - **US Patent Publication 20100174718:** "Indexing for Regular Expressions in Text-Centric Applications"
  - **US Patent Application 12/788,142:** "An Extensible System for Information Extraction in a Data Processing System"

- **Tooling**
  - **US Patent Application 12/269,216**: "User-Guided Regular Expression Learning"
  - **US Patent Application 12/788,407**: "Method for Automatic Refinement of Information Extraction Rules",
  - **US Patent Application 13/117,570**: "A Semantic-Signature-based Method for Contextual Clue Pattern Discovery of Information Extraction"

Information Management

IBM

Introducing Big SQL for BigInsights

IBM's SQL Query Interface for Hadoop

<<Speaker Name Here>>
<<Speaker Title Here>>
<<For questions about this presentation contact Speaker Name speaker@us.ibm.com>

February 8, 2014

Executive Summary

- **What is Big SQL?**
  - Industry-standard SQL query interface for BigInsights data
- **Why Big SQL?**
  - Easy on-ramp to Hadoop for SQL professionals
  - Support familiar SQL tools / applications (via JDBC and ODBC drivers)
- **What SQL operations are supported?**
  - Create tables (and, optionally, HBase indexes)
  - Load data into tables (from local files, distributed files, RDBMSs)
  - Query data (project, restrict, join, union, sub-queries . . . .)
- **What Hadoop-based storage mechanisms are supported?**
  - Hive
  - HBase
  - Distributed file system

Agenda

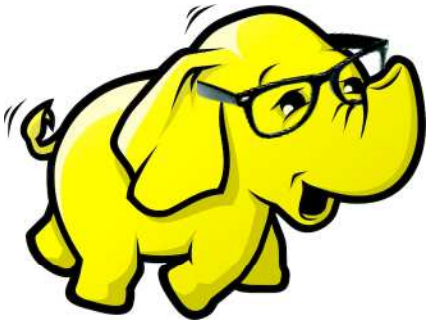
- **Big SQL: motivation and architecture**
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about**

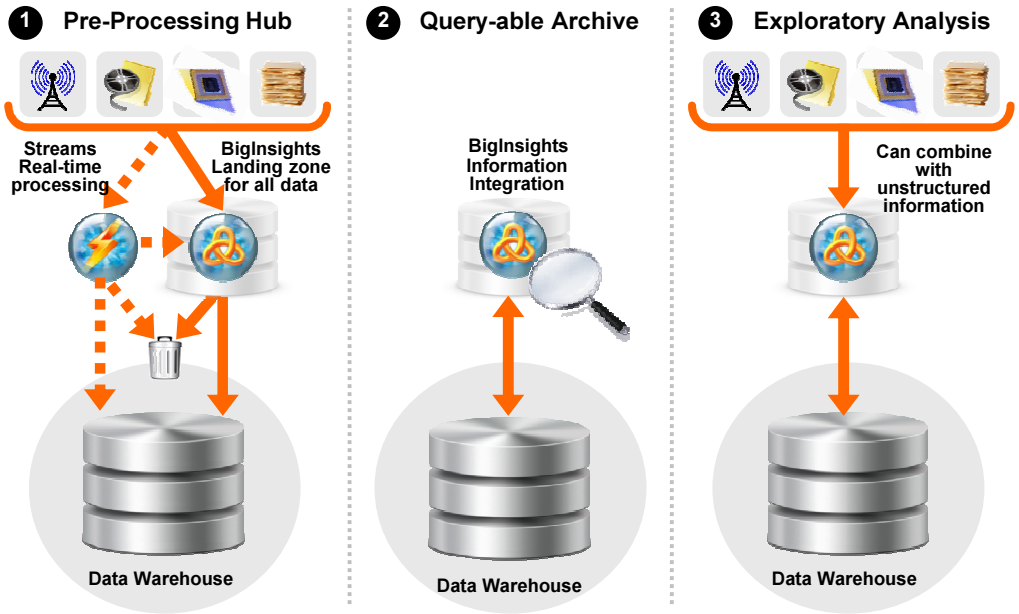
Agenda

- **Big SQL: motivation and architecture** ✓
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about Big SQL**

SQL Access for Hadoop: Why?

- **Data warehouse augmentation is a leading Hadoop use case**



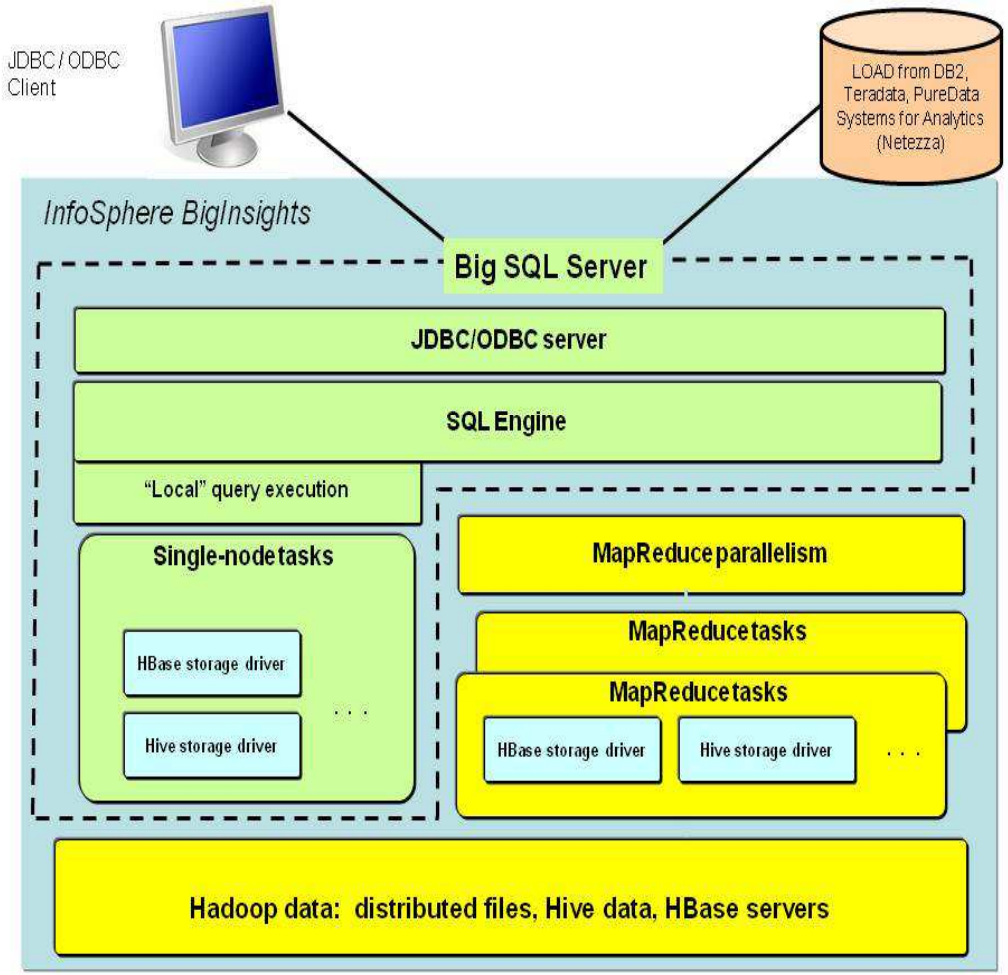


- **Hadoop often perceived as difficult**
  - MapReduce Java API requires programming expertise
  - Unfamiliar languages (such as Pig) also require special skills

- **SQL support opens the data to a much wider audience**
  - Familiar, widely known syntax
  - Common catalog for identifying data and structure

Big SQL Architecture and Feature Overview

- **Standard SQL syntax and data types**
  - Joins, unions, aggregates . . .
  - VARCHAR, decimal, TIMESTAMP, . . .
- **JDBC/ODBC drivers**
  - Prepared statements
  - Cancel support
  - Database metadata API support
  - Secure socket connections (SSL)
- **Optimization**
  - MapReduce parallelism
    or…
  - "Local" access for low-latency queries
- **Varied storage mechanisms appropriate for Hadoop ecosystem**
- **Integration**
  - Eclipse tools
  - DB2, Netezza, Teradata (via LOAD)
  - Cognos Business Intelligence
  - , , ,

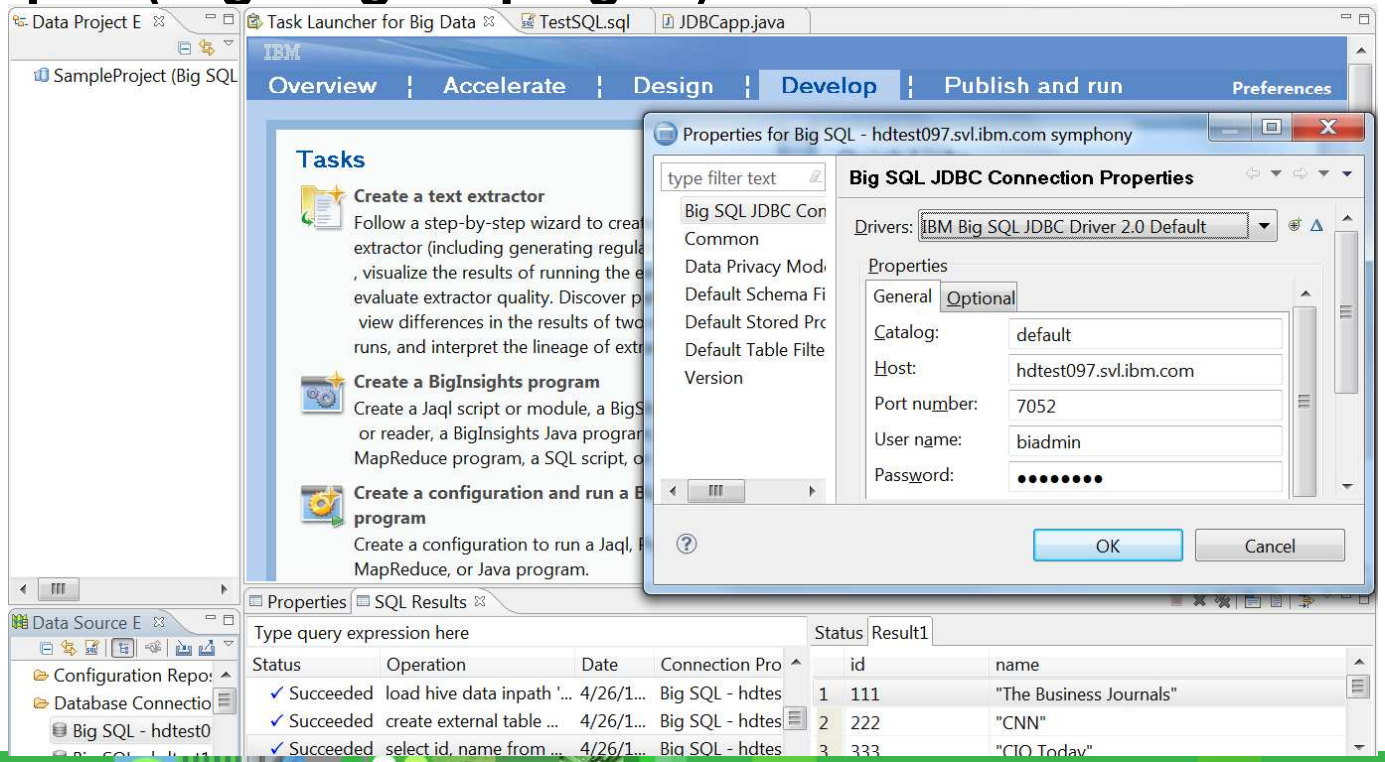


Agenda

- **Big SQL: motivation and architecture**
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about Big SQL**

Invocation options provided with BigInsights

- Command-line interface (JSqsh shell)
- Web-based interface (BigInsights web console)
- Eclipse (BigInsights plug-in)



Creating a Big SQL Table

- **BigSQL supports CREATE TABLE and many data types including varchar, decimals, etc. Non-ISO standard clauses leverage Hadoop ecosystem**

```
CREATE TABLE TPCH.CUSTOMER ( C_CUSTKEY INTEGER, C_NAME VARCHAR(25),
C_ADDRESS VARCHAR(40), C_NATIONKEY INTEGER, C_PHONE CHAR(15), C_ACCTBAL
FLOAT, C_MKTSEGMENT CHAR(10), C_COMMENT VARCHAR(117) )
row format delimited fields terminated by '|'
stored as textfile;
```

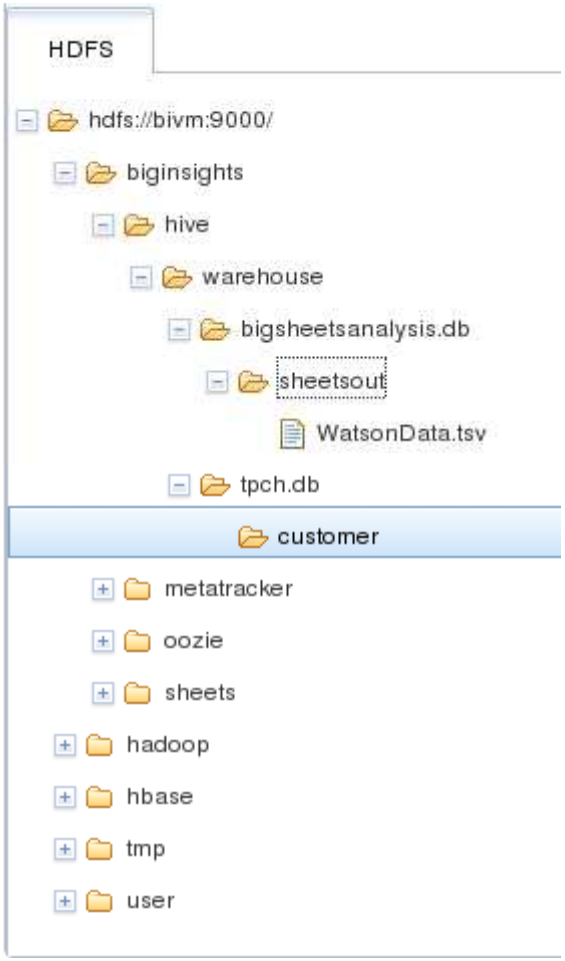
- **Hive does not support data types like varchar and decimal***

```
CREATE TABLE  TPCH.CUSTOMER ( C_CUSTKEY INTEGER, C_NAME VARCHAR(25),
C_ADDRESS VARCHAR(40), C_NATIONKEY INTEGER, C_PHONE CHAR(15),
C_ACCTBAL FLOAT, C_MKTSEGMENT CHAR(10), C_COMMENT VARCHAR(117) )
row format delimited fields terminated by '|'
stored as textfile;
```

*Hive 0.11 added DECIMAL

Results from CREATE TABLE . . .

- **Table**
  - Subdirectory created in warehouse directory
    `/biginsights/hive/warehouse``/`*`tablename`*`/`
  - **External** tables may have their data stored anywhere in the DFS
  - Populated tables contain 1 or more data files
- **Schema (or database)**
  - Tables may be organized by schemas
  - Schema is just a collection of tables
  - Creating a schema creates a subdirectory in the warehouse to hold the tables
    `/biginsights/hive/warehouse/`*`schema`*`.db/`
    `tablename``/`
- **Catalog data (more later)**



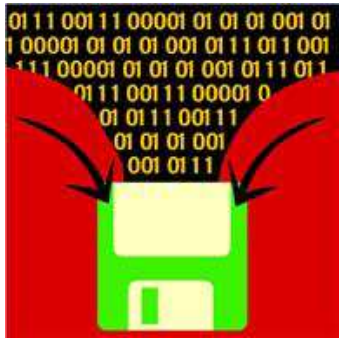
Big SQL Extensions to CREATE TABLE

- **Additional data types: BINARY(N), VARCHAR(N), DECIMAL(P,S)**
- **NULL/NOT NULL indicators**
  - These are advisory only – not enforced
  - Big SQL query re-write software takes advantage of this info
- **Table hints**
  - Certain optimizer hints can be attached to tables
  - Hint will automatically apply when the table is used in a query

```
create table offices
(
officeid int          not null
name      string      not null
)
...
with hints (tablesize='small')
```

- **Explicit syntax for HBase tables (column mappings, column**

Populating Tables



- **Data can be LOADed from . . .**
  - Local file system
  - Distributed file system
  - Remote Netezza, DB2, or Teradata RDBMS
- **Example**

```
CREATE TABLE EMPLOYEE (EMPNO INT, NAME STRING, AGE INT) . . . ;

// Overwrite any existing  data with new data from a local file
LOAD HIVE DATA  LOCAL INPATH '/home/user1/employee.data'  OVERWRITE  INTO TABLE EMPLOYEE;

// Append new data from a file in HDFS to the table
LOAD HIVE DATA INPATH '/user/biadmin/employee.data‘  INTO TABLE EMPLOYEE;
```

- **What LOAD does:**
  - Copies or moves the data, but doesn’t manipulate it
  - Format of the input file must match the format of the table
- **HBase notes:**
  - Similar LOAD syntax (LOAD HBASE ….). Composite keys, indexes, column encoding handled.
  - A single row INSERT may be used against HBase table

Querying data: Overview of SQL Support

- **Projection**

```
SELECT col1, col2  FROM t1
```

- **Restriction**

```
SELECT * FROM t1 WHERE col1 > 5
```

- **Union**

```
SELECT EMPNO FROM EMPLOYEE WHERE WORKDEPT LIKE 'E%'
UNION
SELECT EMPNO FROM ACTIVITIES WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
```

- **Difference (EXCEPT)**

```
(SELECT * FROM T1) EXCEPT ALL (SELECT * FROM T2)
```

- **Intersection**

```
(SELECT * FROM T1) INTERSECT (SELECT * FROM T2)
```

- **Joins**
- **Subqueries**
- **Built-in functions**

SQL Support - Joins

- Big SQL supports both common and ANSI join syntax

```
select ...
from tpch.orders, tpch.lineitem
 where o_orderkey = l_orderkey
```

```
select ...
from tpch.orders join

  tpch.lineitem
  on o_orderkey =
l_orderkey
```

- Hive supports joins via ANSI join syntax only

```
select ...
from tpch.orders, tpch.lineitem
 where o_orderkey = l_orderkey
```

```
select ...
from tpch.orders join

  tpch.lineitem
  on o_orderkey =
l_orderkey
```

SQL Support – Subqueries

- **Big SQL supports subqueries**

```
select c1,
(select count(*) from t2)
from t1
...
```

```
select c1
  from t1
where c2 > (select ...)
```

- **Hive does not support subqueries**

```
select c1,
(select count(*) from t2)
from t1
...
```

```
select c1
  from t1
where c2 > (select ...)
```

SQL Support – Aggregates

year	total_sales	ranked_sales
2006	1495891100.90	1
2005	1159195590.16	2
2007	1117336274.07	3
2004	914352803.72	4

- **Big SQL supports windowed aggregates**

```
SELECT EXTRACT(YEAR FROM CAST(CAST (order_day_key AS varchar(100)) AS
timestamp)) AS year,
SUM (sale_total) AS total_sales,
RANK () OVER (ORDER BY SUM (sale_total) DESC) AS ranked_sales
FROM gosalesdw.sls_sales_fact
GROUP BY EXTRACT(YEAR FROM CAST(CAST (order_day_key AS varchar(100))
AS timestamp))
```

- **Hive does not support windowed aggregates**

```
SELECT EXTRACT(YEAR FROM CAST(CAST (order_day_key AS varchar(100)) AS
timestamp)) AS year,
SUM (sale_total) AS total_sales,
RANK () OVER (ORDER BY SUM (sale_total) DESC) AS ranked_sales
FROM gosalesdw.sls_sales_fact
GROUP BY EXTRACT(YEAR FROM CAST(CAST (order_day_key AS varchar(100))
AS timestamp))
```

SQL Support – Functions (partial list)

- **Wide variety of built-in functions**
  - Numeric

abs	ceil	floor	ln	log10
mod	power	sqrt	sign	width_bucket

  - Trigonometric

cos	sin	tan	acos	asin
atan	cosh	sinh	tanh	

  - Date

_add_days	_add_months	_add_years	localtimestamp	_age
_day_of_week	_day_of_year	_week_of_year	_days_between	_months_between
_years_between	_ymdint_between	_first_of_month	_last_of_month	extract

  - String

char_length	bit_length	octet_length	upper	lower
substring	position	index	translate	trim
json_get_object				

  - Aggregates, etc.

Catalog Tables (HCatalog)

```
[localhost][foo] 1> select * from syscat.tables where tablename='users';
+------------+-----------+
| schemaname | tablename |
+------------+-----------+
| default    | users     |
+------------+-----------+
1 row in results(first row: 0.14s; total: 0.15s)
```

```
[localhost][foo] 1> select * from syscat.columns where tablename='users';

+------------+-----------+-----------+--------+-----------+-------+
| schemaname | tablename | name      | type   | precision | scale |
+------------+-----------+-----------+--------+-----------+-------+
| default    | users     | id        | INT    |        10 |     0 |
| default    | users     | office_id | INT    |        10 |     0 |
| default    | users     | name      | STRING |         0 |     0 |
| default    | users     | children  | ARRAY  |         0 |     0 |
+------------+-----------+-----------+--------+-----------+-------+
4 rows in results(first row: 0.19s; total: 0.21s)
```

Other BigInsights catalog tables track index and schema information

Using Existing Standard SQL Tools: Eclipse

BigInsights - .sqlxeditor_project/Script10.sql - Eclipse SDK

File Edit Navigate Search Project Run Script Window Help

BigInsights Servers

*Script10.sql

```
SELECT
  L_RETURNFLAG,
  L_LINESTATUS,
  SUM(L_QUANTITY) AS SUM_QTY,
  SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS SUM_DISC_PRICE,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)*(1+L_TAX)) AS SUM_CHARGE,
  AVG(L_QUANTITY) AS AVG_QTY,
  AVG(L_EXTENDEDPRICE) AS AVG_PRICE,
  AVG(L_DISCOUNT) AS AVG_DISC,
  COUNT(*) AS COUNT_ORDER
FROM LINEITEM
WHERE L_SHIPDATE <= '1998-09-02'
GROUP BY L_RETURNFLAG, L_LINESTATUS
ORDER BY L_RETURNFLAG, L_LINESTATUS
```

Problems | Console | Call Hierarchy | Search

To display the call hierarchy, select one or more methods, c

Help | Data Source Explorer

- Big SQL - bdvm072.svl.ibm.com
- BigSQL - 170.224.193.37 (IBM Bi
  - default
    - Schemas
      - default
      - gosales
      - gosalesdw
      - gosaleshr
      - gosalesmr
      - gosalesrt

Properties for BigSQL - 170.224.193.37

type filter text

- Big SQL JDBC Connec
- Common
- Default Schema Filter
- Default Stored Proce
- Default Table Filter
- Version

Big SQL JDBC Connection Properties

Drivers: IBM Big SQL JDBC Driver 2.0 Default

Properties

General | Optional

Catalog:	default
Host:	170.224.193.37
Port number:	7052
User name:	biadmin
Password:	•••••••

☑ Save password

Connection URL: jdbc:bigsql://170.224.193.37:7052/default

Test Connection

OK | Cancel

Using Existing Standard SQL Tools: SQuirreL SQL

SQuirreL SQL Client Version 3.4.0

File Drivers Aliases Plugins Session Windows Help

Connect to: BigSQL vm... Active Session: 1 - BigSQL vm072 () as biadmin

Aliases: BigSQL vm072

1 - BigSQL vm072 () as biadmin

Objects | SQL

SELECT L_RETURNFLAG, L_LINESTATUS, SUM(L_QUANTITY) AS SUM_QTY, SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE, SUM(L_EXTENDE... Limit rows: 100

```
  SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS SUM_DISC_PRICE,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)*(1+L_TAX)) AS SUM_CHARGE,
  AVG(L_QUANTITY) AS AVG_QTY,
  AVG(L_EXTENDEDPRICE) AS AVG_PRICE,
  AVG(L_DISCOUNT) AS AVG_DISC,
  COUNT(*) AS COUNT_ORDER
FROM tpch.LINEITEM
WHERE L_SHIPDATE <= '1998-09-02'
GROUP BY L_RETURNFLAG, L_LINESTATUS
ORDER BY L_RETURNFLAG, L_LINESTATUS;
```

select count(*) | SELECT L_RETURN

Rows 4: SELECT L_RETURNFLAG, L_LINESTATUS, SUM(L_QUANTITY) AS SUM_QTY, SUM(L_E

Results | MetaData | Info | Overview

l_returnflag	l_linestatus	SUM_QTY	SUM_BASE_PRICE	S
A	F	380,456	532,348,211.54083	505,82
N	F	8,971	12,384,801.3941	11,798
N	O	742,802	1,041,502,841.74463	989,73
R	F	381,449	534,594,445.51434	507,99

Add Alias

Add Alias

Name: BigSQL on bdvm072.svl.ibm.com

Driver: ✓ IBM Big SQL JDBC Driver 2.0 Default | New

URL: jdbc:bigsql://bdvm072.svl.ibm.com:7052/default

User Name: biadmin

Password: •••••••

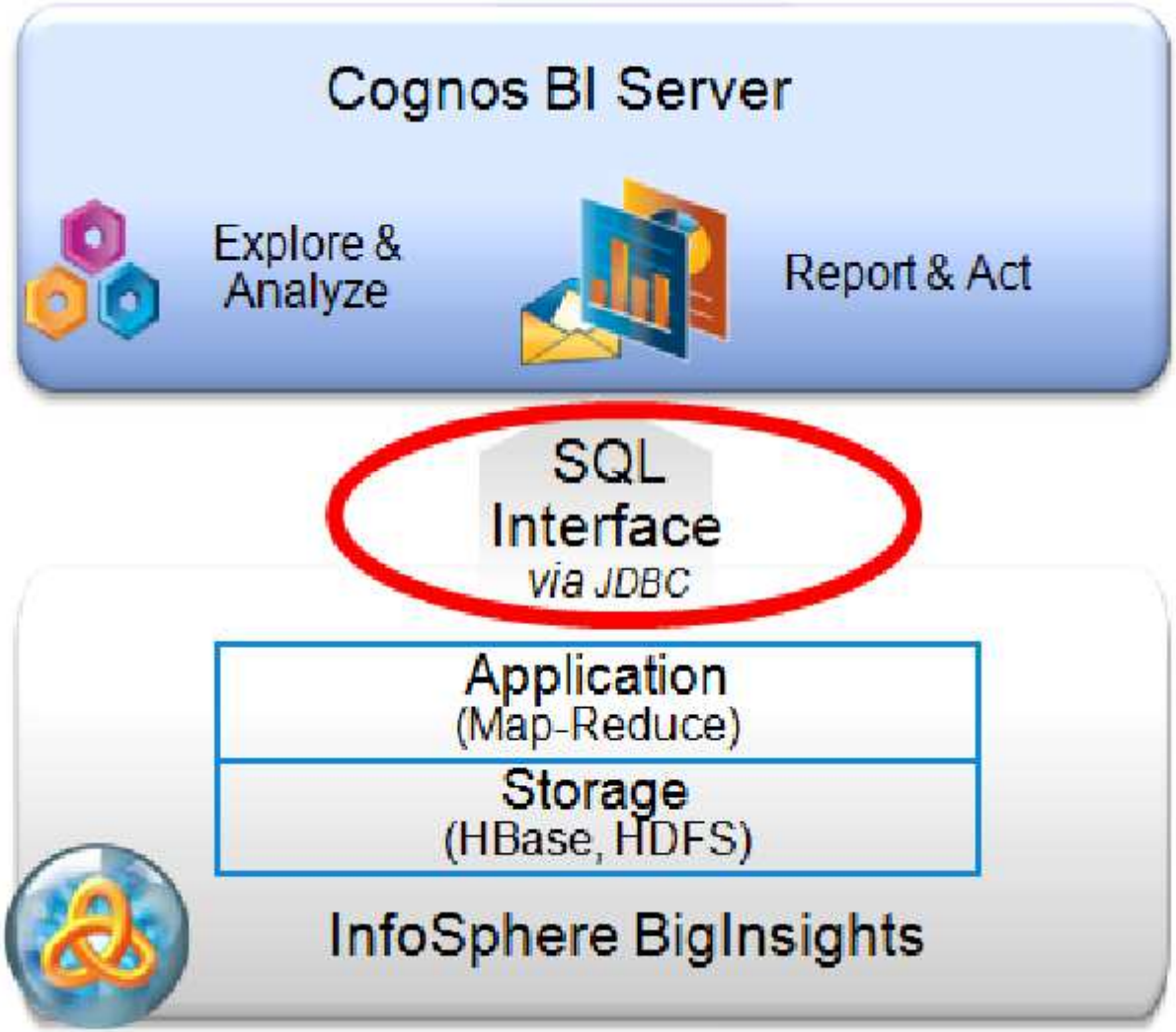
☐ Auto logon ☐ Connect at Startup

Properties

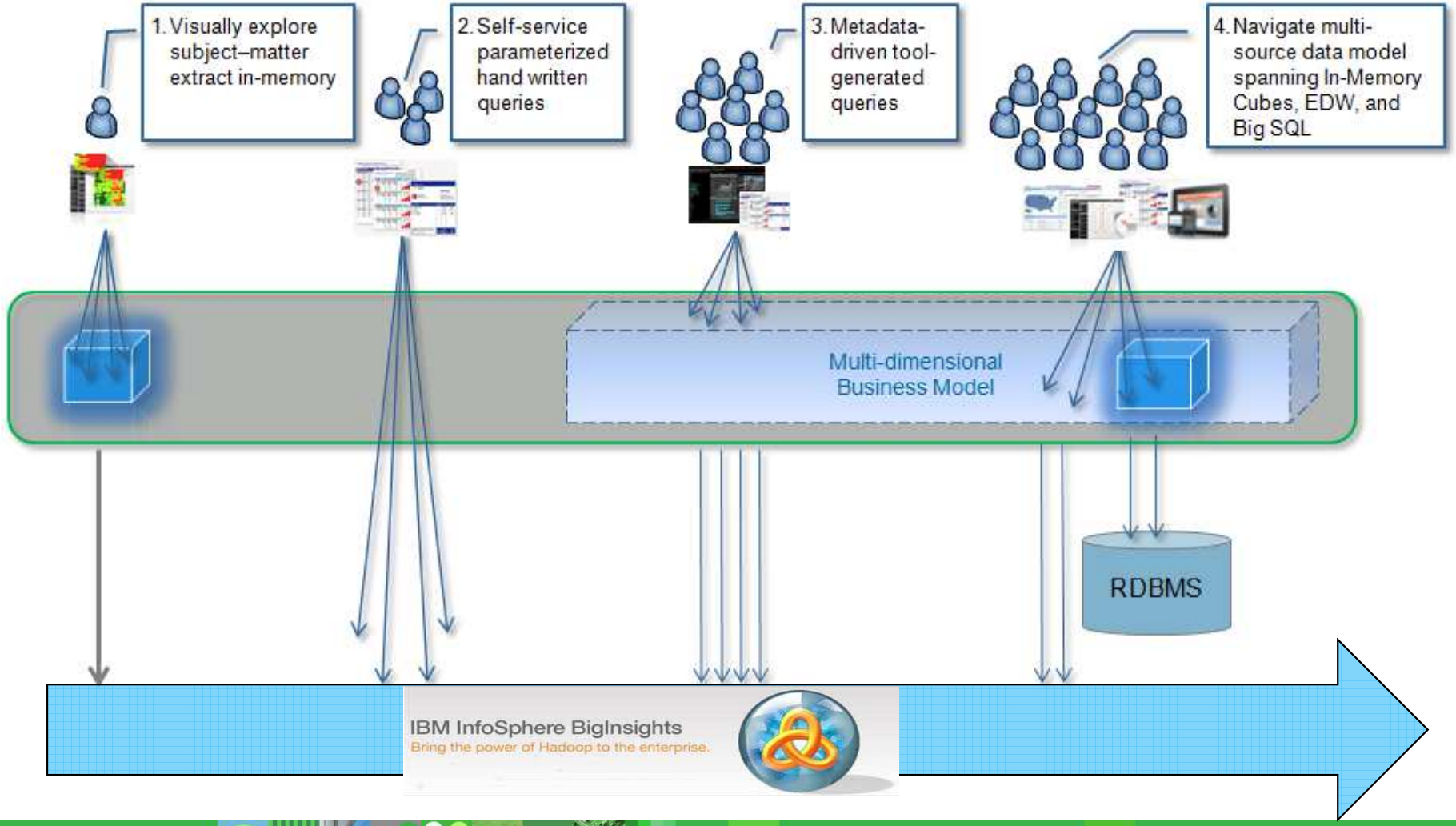
Warning - Passwords are saved in clear text

OK | Close | Test

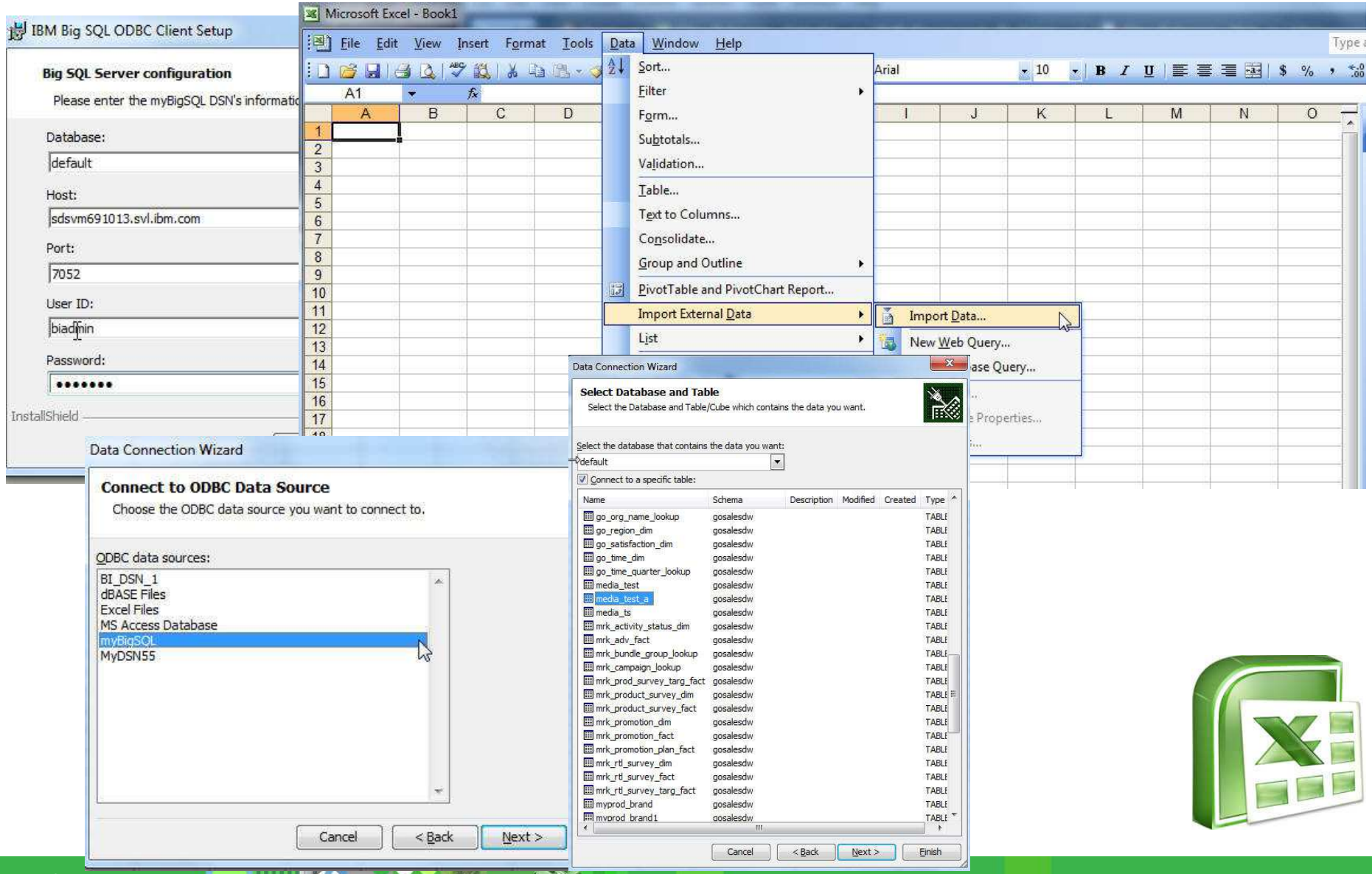
Cognos Business Intelligence



MicroStrategy use of Big SQL



MS Excel: Big SQL integration via ODBC



A word about . . . SerDes

- **Custom serializers / deserializers (SerDes)**
  - Read / write complex or “unusual” data formats (e.g., JSON)
  - Commonly used with Hive, HBase
  - Developed by user or available from open source community

- **Using SerDes with Big SQL**
  - Add the SerDe .jar file to `$BIGSQL_HOME/userlib` and `$HIVE_HOME/lib`
  - Stop / restart Big SQL service
  - Specify SerDe class name (not .jar file name) when creating table

- **Example**

```
/* Create a table for JSON data. Use open source hive-json-serde-0.2.jar SerDe */
create table socialmedia-json (Country String, FeedInfo String, . . . )
row format serde 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
stored as textfile;

load hive data inpath '</hdfs_path>/WatsonBlogsData.json' overwrite into table
  socialmedia-json;

select * from socialmedia-json;
```

Sample JSON input for previous example

```
[biadmin@bdvm327 twitter]$ cat WatsonNewsBlogsData.json|more

[{"PostSize":5775,"ThreadId":"4f129a8be","Crawled":"2012-01-15 09:21:15","FeedInfo":"{\"Tit
le\":\"www.ibm.comnews\",\"Id\":\"44032787\",\"ExtKey\":\"879cd3257c296614160914c3d96f9b85\
",\"Url\":\"http://www-03.ibm.com\"}","Published":"2012-01-15 09:21:15","Url":"http://www.i
bm.com/innovation/us/watson/?lnk=ftpl","Country":"US","SubjectHtml":"<Keyword>IBM<\/Keyword
> - <Keyword>Watson<\/Keyword>","Inserted":"2012-05-29 00:52:57","Language":"English","Text
Html":"<![CDATA[<Keyword>IBM<\/Keyword> - <Keyword>Watson<\/Keyword>\n\n      Call to find
out how Watson's capabilities could benefit your business.\n      1-800-426-7630\n    \nRel
ated content\nDesigning the Computer for a Smarter Planet\nThere\u2019s an enormous amount
of science included when <Keyword>Watson<\/Keyword> answers a single Jeopardy! question, ho
w does it all work together?\nExplore <Keyword>Watson<\/Keyword>\nBeyond Jeopardy! The Busi
ness Implications of <Keyword>Watson<\/Keyword>\n<Keyword>IBM<\/Keyword> <Keyword>Watson<\/
Keyword> passed its first test onÂ Jeopardy!Â in February 2011, but the real test will be i
n
applying the underlying systems, data management and analytics technology in business and a
cross different industries. Watch the webcast now and learn about the present and future bu
siness implications of Deep QA and the other technologies behind <Keyword>Watson<\/Keyword>
 from David Ferrucci and other <Keyword>IBM<\/Keyword> executives.\nRegister now\nBetter Bu
```

JSON-based social media data to load into Big SQL Table *socialmedia-json* defined with SerDe

Sample Big SQL query output for JSON data

authorinfo	country	crawled	feedinfo	id	inserted	published
{"Nick":"","Id":"","Name":"","Url":""}	US	2012-04-11 03:31:47	{"Title":"www.ibm.comnews","Id":"4403278 7","ExtKey":"879cd32 57c2966141 60914c3d96 f9b85","Ur l":"http:/ /www-03.ib m.com"}	31859312 32	2012-05-23 20:18:08	2012-04-11 03:31:47
{"Nick":"","Id":"","Name":"","Url":""}	US	2012-02-22 23:50:14	{"Title":"www.ibm.comnews","Id":"4403278 7","ExtKey":"879cd32 57c2966141 60914c3d96 f9b85","Ur l":"http:/ /www-03.ib m.com"}	32535948 48	2012-05-26 05:21:06	2012-02-22 23:50:14

Sample output: Select * from socialmedia-json

A word about . . . performance

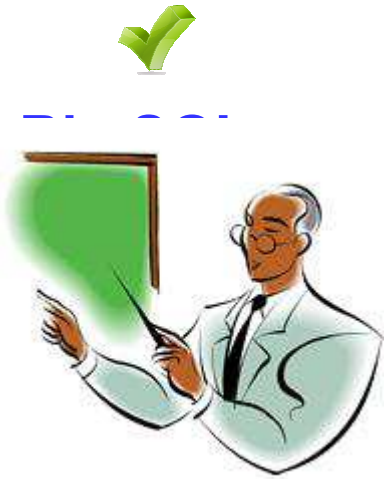
- **Tuning options**
  - Table design (e.g., storage formats for Hive, key & column family definitions for HBase)
  - Hints in queries
  - Hints in table definition
  - Secondary indexes (HBase tables only)
  - MapReduce job properties
- **Query hints provided in comments: `/*+ name=value [, …] +*/`**

```
select * from foo /*+ accessmode='local' +*/ where c1 < 1000;
```

- **Access mode hint**
  - **Causes query to be executed in the Big SQL server**
  - **HBase indexed queries can return extremely rapidly**
  - **Local access can be forced on for your entire session**

Agenda

- **Big SQL: motivation and architecture**
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about**



Big SQL – what RDBMS experts should know

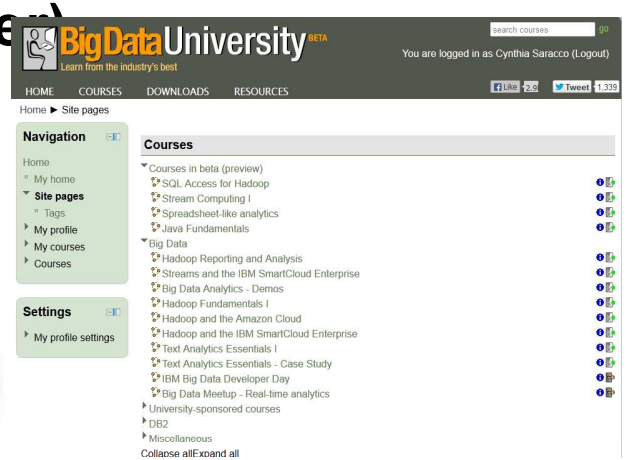
- **Big SQL provides industry-standard query support for Hadoop-based storage managers**
  - Exploits Hadoop environment
  - Includes Hadoop-specific extensions
  - Introduces Hadoop-specific concepts
  - Copes with "unconventional" data structures and formats (e.g., JSON) via SerDes, other features

- **RDBMS = more than query & storage management**
  - Transaction management
  - Views
  - Stored procedures
  - INSERT / UPDATE / DELETE
  - GRANT / REVOKE
  - 3GL language support (e.g., COBOL)
  - Rich catalog statistics and decades of cost-based optimization development

- **Bottom line: Big SQL provides SQL experts with on-ramp to Hadoop, but doesn't turn Hadoop into one big relational database**

Want to learn more?

- **Big SQL tutorial (product Information Center)**
- **Videos , articles, downloads, etc.**
  - Technical portal at http://tinyurl.com/biginsights



InfoSphere BigInsights Tutorials

Manage

Within minutes, dive into the world of big data with robust, browser-based control.

Import

Collect and import data for exploration and analysis that helps you make sense of seemingly unrelated data.

Analyze

Delve into BigSheets, an intuitive spreadsheet-like tool, to create analytic queries without any previous programming experience.

Develop

Easily develop your first big data application by using the InfoSphere BigInsights Eclipse plugin.

Query

Quickly master the intricacies of SQL queries for Hadoop with IBM Big SQL.

Extract

Discover the power of Text Analytics by creating extractors to derive valuable insights from text documents.

YouTube

BigInsights Technical Enablement Wiki

IBM InfoSphere BigInsights

Get up to speed on InfoSphere BigInsights, IBM's software platform designed to help firms store, manage, and analyze "big data".

Technical materials

- Articles, white papers, and books
- BigInsights InfoCenter

Videos and Demos

- Video guide

Downloads

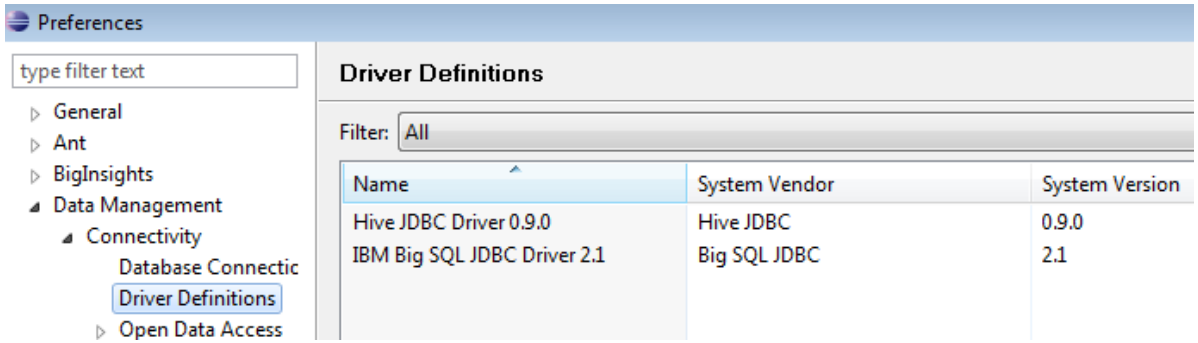
- BigInsights Basic Edition (free)
- Karmasphere Studio Community Edition Virtual Appliance with BigInsights (free)
- Fix packs for BigInsights Enterprise Edition (licensed)

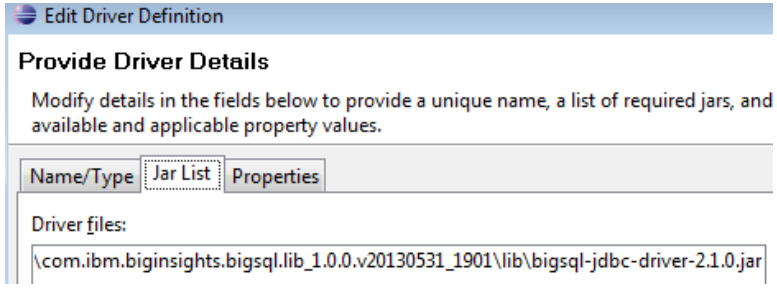
Discussion Forum

- BigInsights forum on IBM developerWorks

Big SQL JDBC driver definition (Eclipse)

- A JDBC driver for Big SQL 2.1 is created automatically when BigInsights server is added
- New driver can be added and customized from Preference.
- Window→ Preferences→ Data Management → Connectivity→ Driver Definition → "Add…"





JSqsh – Big SQL's CLI

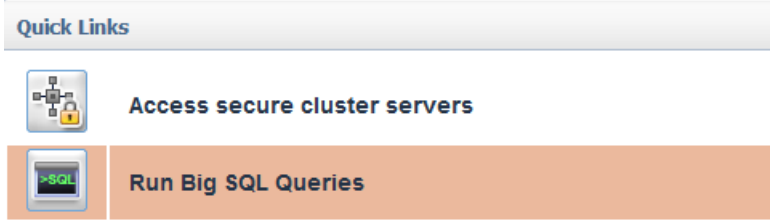
- **JSqsh ("jay-skwish" – Java SQL Shell)**
  - Open source command line JDBC client (http://jsqsh.wiki.sourceforge.net)
  - Works with any JDBC driver, not just Big SQL

- **It can be started with**

```
$ $BIGSQL_HOME/bin/jsqsh --driver=bigsql --user=biadmin --password=biadmin
JSqsh Release 1.5-ibm, Copyright (C) 2007-2013, Scott C. Gray
Type \help for available help topics. Using JLine.
[localhost][biadmin] 1> select * from syscat.tables;
+------------+--------------+
| schemaname | tablename    |
+------------+--------------+
| syscat     | columns      |
| syscat     | tables       |
| syscat     | schemas      |
| syscat     | indexcolumns |
| system     | dual         |
| system     | integers     |
+------------+--------------+
```

BigInsights Web Console

- In Quick Links, select to run Big SQL queries from the console



- Type in query, or cut and paste from SQL script. Hit Run.

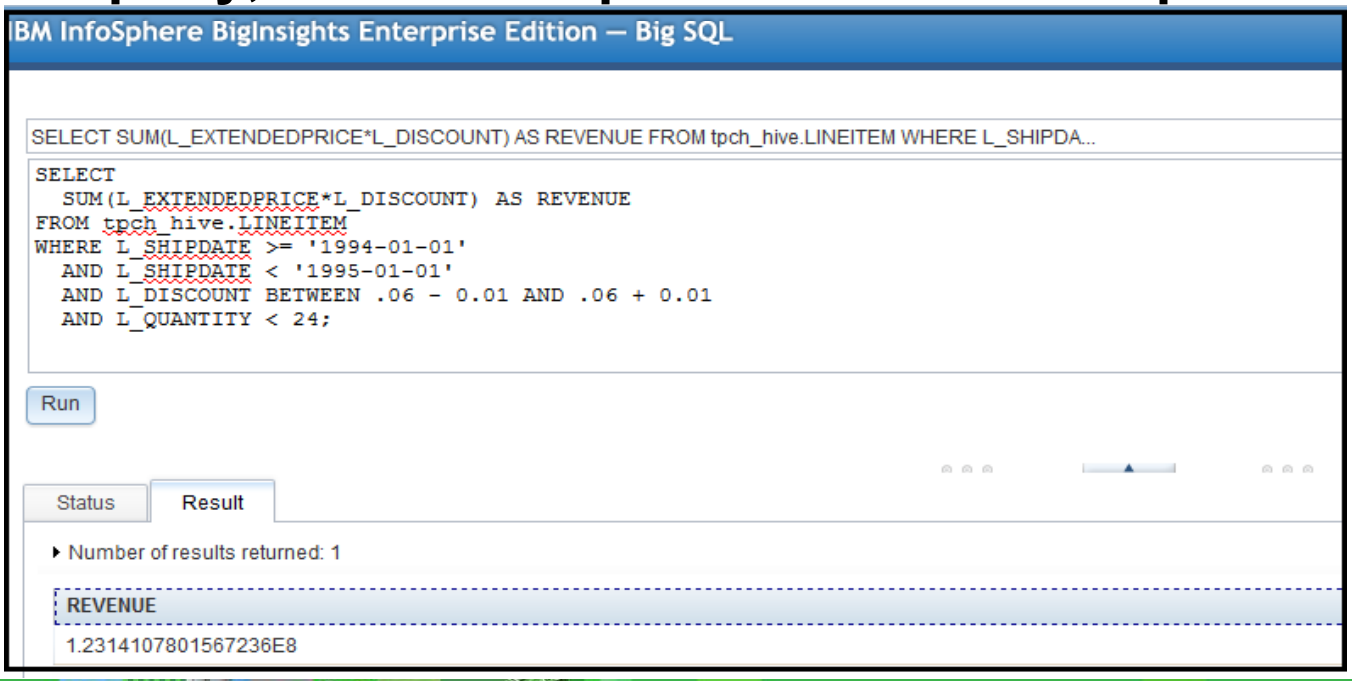
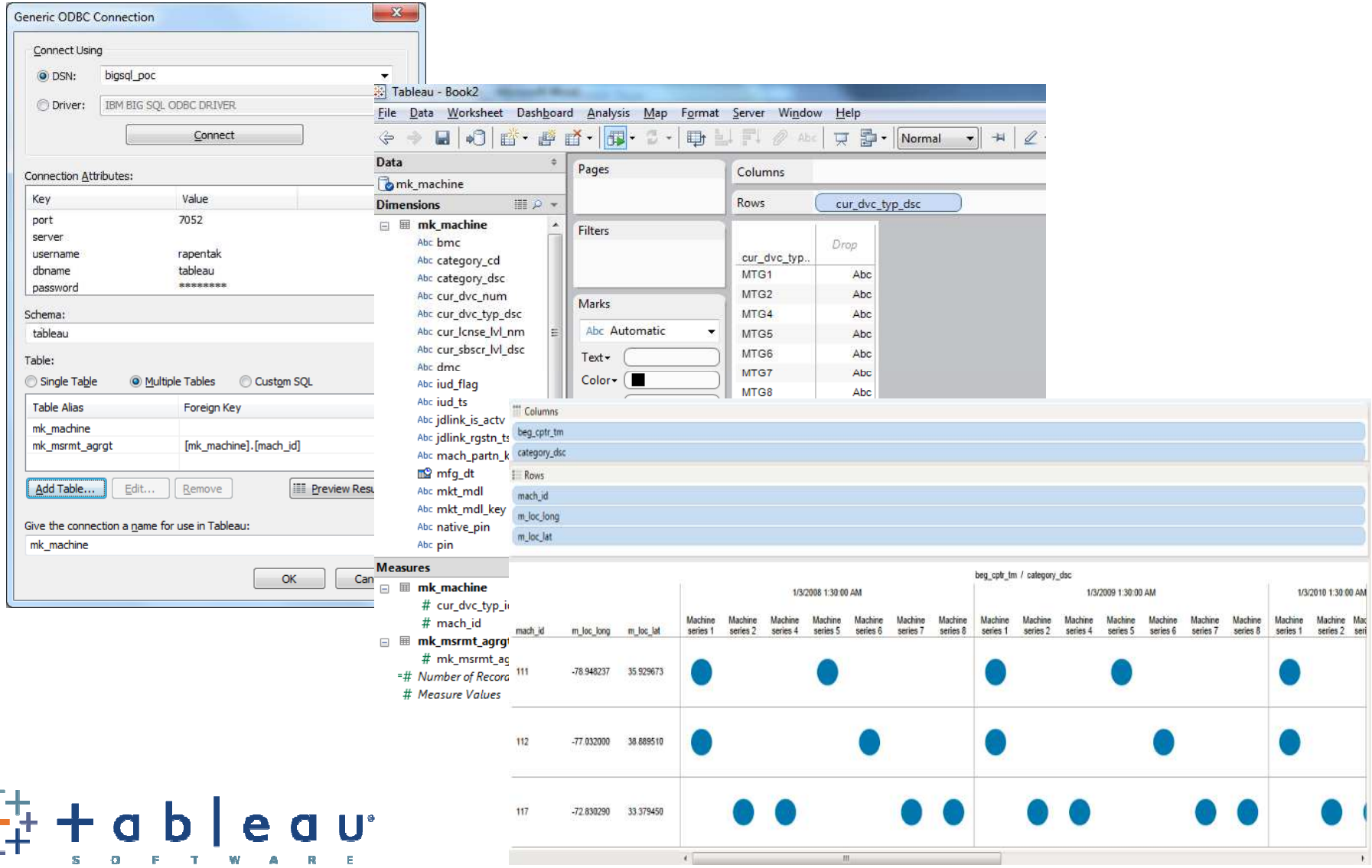


Tableau: Big SQL integration via ODBC



Lotus Symphony: Big SQL integration via JDB

Spreadsheet and Database - IBM Lotus Symphony

File View Tools Window Help

New | New Spreadsheet 3

B7 | ETHIOPIA

	A	B	C	D
1	n_nationkey	n_name	n_regionkey	n_comment
2	0	ALGERIA	0	final accounts wake quickly. special reques
3	1	ARGENTINA	1	idly final instructions cajole stealthily. regular instructions wake carefully blith
4	2	BRAZIL	1	always pending pinto beans sleep sil
5	3	CANADA	1	foxes among the bold requests
6	4	EGYPT	4	pending accounts haggle furiously. furiously bold accounts detect. platelets a
7	5	ETHIOPIA	0	fluffily ruthless requests integrate fluffily. pending ideas wake blithely acco
8	6	FRANCE	3	even requests detect near the pendin
9	7	GERMANY	3	blithely ir
10	8	INDIA	2	ironic pac
11	9	INDONESIA	2	unusual e
12	10	IRAN	4	blithely e
13	11	IRAQ	4	express,
14	12	JAPAN	2	blithely fi
15	13	JORDAN	4	blithe, ex
16	14	KENYA	0	ironic req
17	15	MOROCCO	0	ideas acc
18	16	MOZAMBIQUE	0	ironic cou
19	17	PERU	1	final, fina
20	18	CHINA	2	bold acco
21	19	ROMANIA	3	deposits
22	20	SAUDI ARABIA	4	fluffily fin
23	21	VIETNAM	2	doggedly
24	22	RUSSIA	3	slowly pe
25	23	UNITED KINGDOM	3	fluffily re
26	24	UNITED STATES	1	blithely regular deposits serve furiously blithely regular warthogs! slyly fi
27				
28				
29				
30				
31				
32				
33				

Sheet1 Sheet2 Sheet3

Modify Aliase:bigsql

Modify Aliase:bigsql

Name: bigsql

Loaded Driver: Big SQL JDBC Driver

URL: jdbc:bigsql://hdtest102.svl.ibm.com:7052/defau

User Name: biadmin

Password: *******

OK Close Test

My Widgets

Spreadsheet and Da...

DB Drivers

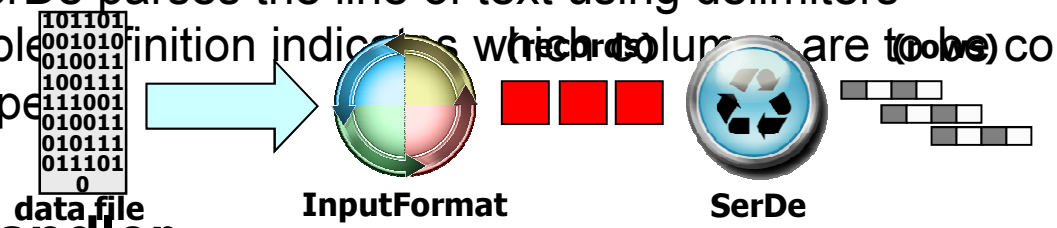
Data Sources

bigsql

Import

More about SerDes

- **SerDe (Serializer/Deserializer)**
  - A Hive concept
  - A java class responsible for converting a **record** produced by an InputFormat into a Hive **row**, based upon the table definition
  - The Hive LazySimpleSerDe
    - Expects **records** from a TextInputFormat – a record is just a single line of text
    - The SerDe parses the line of text using delimiters
    - The table definition indicates which columns are to be converted to which data type



- **Storage Handler**
  - A Hive concept
  - A java class that interacts with an external data source
  - Contains an InputFormat and SerDe to communicate data
  - Is presented with query projection and predicates to optimize data access
  - HBase is currently the only Storage Handler

Hive at a Glance

- **Open source data warehouse framework for Hadoop**
- **Data stored in DFS files, but programmers create / query tables**
- **Provides SQL-like interface (Hive Query Language, HQL)**
  - Language constructs cover a subset of commercial SQL
  - Queries run as MapReduce jobs under the covers
  - Programmers can create custom Mappers, Reducers
- **From Hive wiki:**

  Hive is not designed for OLTP workloads and does not offer real-time queries or row-level updates. It is best used for batch jobs over large sets of append-only data (like web logs)

Hive storage

- **Warehouse directory in DFS**
  - Specified by "*hive.metastore.warehouse.dir*" in hive-site.xml
  - /biginsights/hive/warehouse the default location for BigInsights

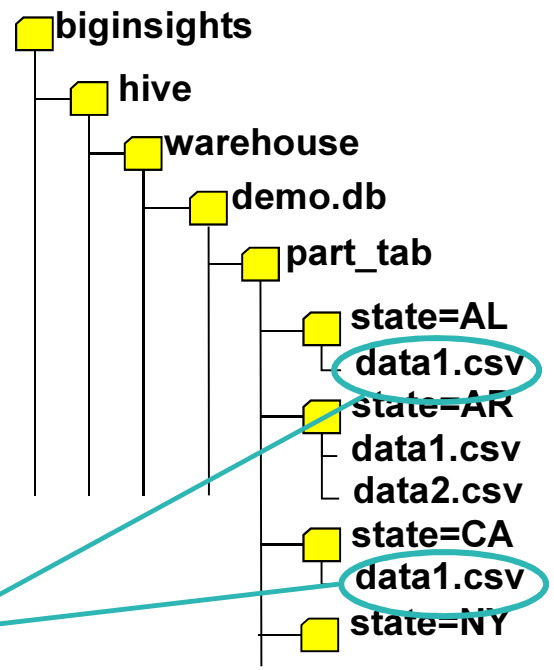
- **One can think tables, partitions and buckets as directories, subdirectories and files respectively**

Hive Entity	Sample	Sample location in DSF
database	**test**	**$WH/test.db**
table	**T**	**$WH[/test.db]/T**
partition	**date='01012013'**	**$WH/T/date=01012013**
bucketing column	**userid**	**$WH/T/date=01012013/000000_0** **… …** **$WH/T/date=01012013/000032_0**

Partitioned Tables

- All tables except HBase can be partitioned
- Partitioning is on one or more columns
- Each unique value becomes a partition
- Query predicates can be used to eliminated scanned partitions

```
CREATE TABLE demo.sales (
        part_id int,
     part_name string,
          qty int,
        cost double
          )
    PARTITIONED BY (
       state char(2)
          )
  ROW FORMAT DELIMITED
 FIELDS TERMINATED BY '|';
```




```
select *
from demo.sales
where state in ('AL', 'CA');
```

Why Another Data Warehousing System? A perspective from Facebook

- **Problem: Data, data and more data**
  - 200GB per day in March 2008
  - 2+TB(compressed) raw data per day today
- **The Hadoop Experiment**
  - Superior to availability/scalability/manageability of commercial DBs
  - Efficiency not that great, but throw more hardware
  - Partial Availability/resilience/scale more important than ACID
- **Problem: Programmability and Metadata**
  - Map-reduce hard to program (users know sql/bash/python)
  - Need to publish data in well known schemas
- **Solution: HIVE**

Excerpt from 2008 presentation by Facebook

HBase at a Glance

- **Open source key-value data store for Hadoop**
  - Based on Google's Bigtable paper [2006]
  - Implemented as a sparse, consistent, distributed, multi-dimensional, persistent, sorted map
  - Key and value are byte arrays

- **Strengths**
  - Efficient read/write access using row key, small range scan
  - Very good for "sparse data" (no fixed columns)
  - Highly scalable
  - Rich set of Java APIs and extensible frameworks

- **Different from relational databases**
  - No types: all data is stored as bytes
  - No schema: Rows can have different set of columns
  - No native SQL support
  - No multi-row transactions
  - Not optimized for N-way joins scanning large portions of data set

HBase Data Model

- **Table**
  - Contains column-families
- **Column family**
  - Logical and physical grouping of columns
- **Column**
  - Exists only when inserted
  - Can have multiple versions
  - Each row can have different set of columns
  - Each column identified by it's key
- **Row key**
  - Implicit primary key
  - Used for storing ordered rows
  - Efficient queries using row key

HBTABLE	
Row key	**Value**
11111	cf_data: {'cq_name': 'name1', 'cq_val': 1111} cf_info: {'cq_desc': 'desc11111'}
22222	cf_data: {'cq_name': 'name2', 'cq_val': 2013 @ ts = 2013, 'cq_val': 2012 @ ts = 2012 }

HFile

11111 cf_data cq_name name1 @ ts1
11111 cf_data cq_val 1111 @ ts1
22222 cf_data cq_name name2 @ ts1
22222 cf_data cq_val 2013 @ ts1
22222 cf_data cq_val 2012 @ ts 2

HFile

11111 cf_info cq_desc desc11111 @ ts1

HBase Support

- **Robust HBase support is a major Big SQL focus**
- **HBase is different than most other data sources in Hadoop**
  - Client/server database
    - Fetching rows/columns requires a network hop
  - Efficiently querying HBase requires pushing as much to the server(s) as possible
    - Pushing down query predicates as filters to region servers
    - Fetching only columns needed by the query
  - All HBase tables are ordered and accessed by primary key
    - Big SQL leverages this

Creating HBase Tables

- **Hive syntax for HBase tables is cumbersome**
  - Difficult to read
  - Cannot express composite keys and columns

- **Big SQL provides explicit syntax for defining tables in Hive**

```
create hbase table sales(
prod_id     int    not null,
sales_date  int    not null,
quantity    int    not null,
price       double not null
)
column mapping
(
key mapped by (prod_id, sales_date) encoding binary,
cf1:sales_data mapped by
(quantity, price) separator '|' encoding string
)
column family options
cf1 compression gz, bloom filter none, in memory
hbase table name 'PROD_SALES'
default encoding binary
default column family options compression none
```

HBase Hints

- **rowcachesize (default=2000)**
  - Used as scan cache setting
  - Also used to determine number of get requests to batch in index lookups
- **colbatchsize (default=100)**
- **useindex ('false' to avoid index usage)**

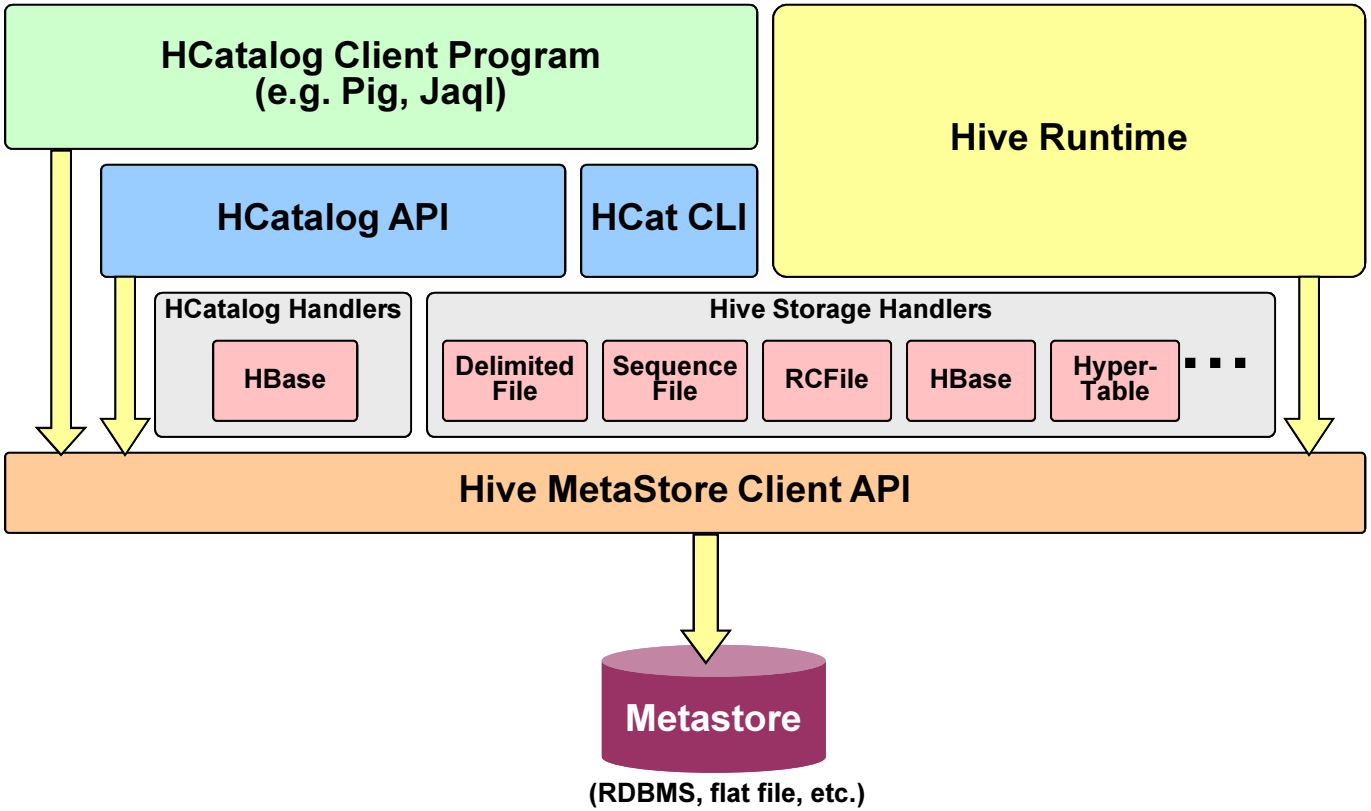
```
select o_orderkey from orders /*+ rowcachesize=10000 +*/ where o_custkey>5000
g  discard
1450136 rows in results(first row: 22.67s; total: 27.46s)

        HBase scan details:{... , caching=10000, ...}
```

Log

- **rowcachesize can also be set using the set command:**
  - set hbase.client.scanner.caching=10000;

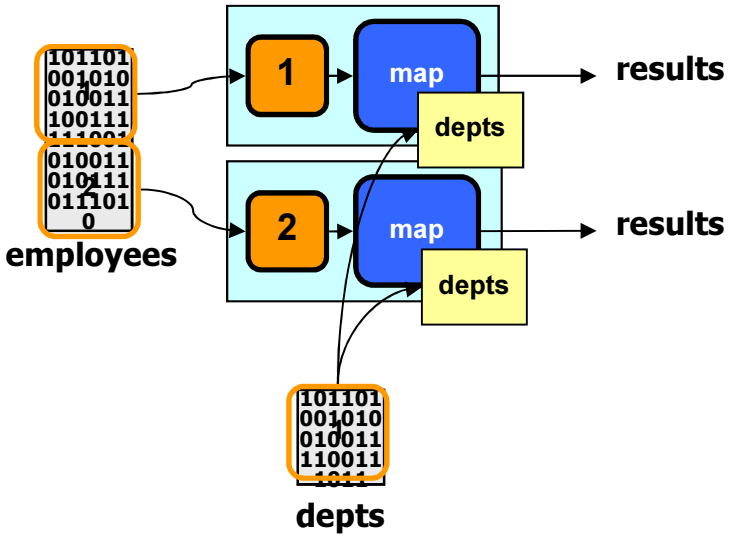
HCatalog

- API's to read/write data directly from Hive "tables"
- Tools for manipulating metadata (create/drop/alter tables)
- Provide enhanced HBase support for use cases beyond Hive



Dynamic Query Optimization

- **During query execution Big SQL dynamically re-evaluates its options** `SELECT c1, c2 FROM T1`
  - Queries that cannot be assisted by MapReduce run in the server
  - If a given step (e.g. GROUP or SORT) involves only "small" data, the step is executed in the server
  - If all tables are small, the whole query will be run in-server
  - If one table is large and one or more are small, a memory (hash) join is performed



Performance and Tuning

- **Big SQL will dynamically adjust query strategies**
  - Execute steps in the server
  - Automatically choose memory joins vs. redistribution join
- **Server config settings can adjust how decisions are made**
  - And can be adjust at the session level using `SET` command
- **Query join order (in the FROM clause) can impact performance**
  - Order you provide is honored
  - Most selective data sets should be first
- **Query hints and be used to fine-tune performance**
  - Table hints can adjust join strategy for specific tables
  - Table access hints and fine-tune access to specific data sources (e.g. HBase row fetch sizes)
  - Specific joins can be forced to execute locally (in-server) vs. MapReduce

Information Management

IBM

Developing, publishing and deploying your first Big SQL application with InfoSphere BigInsights

< insert your name here>

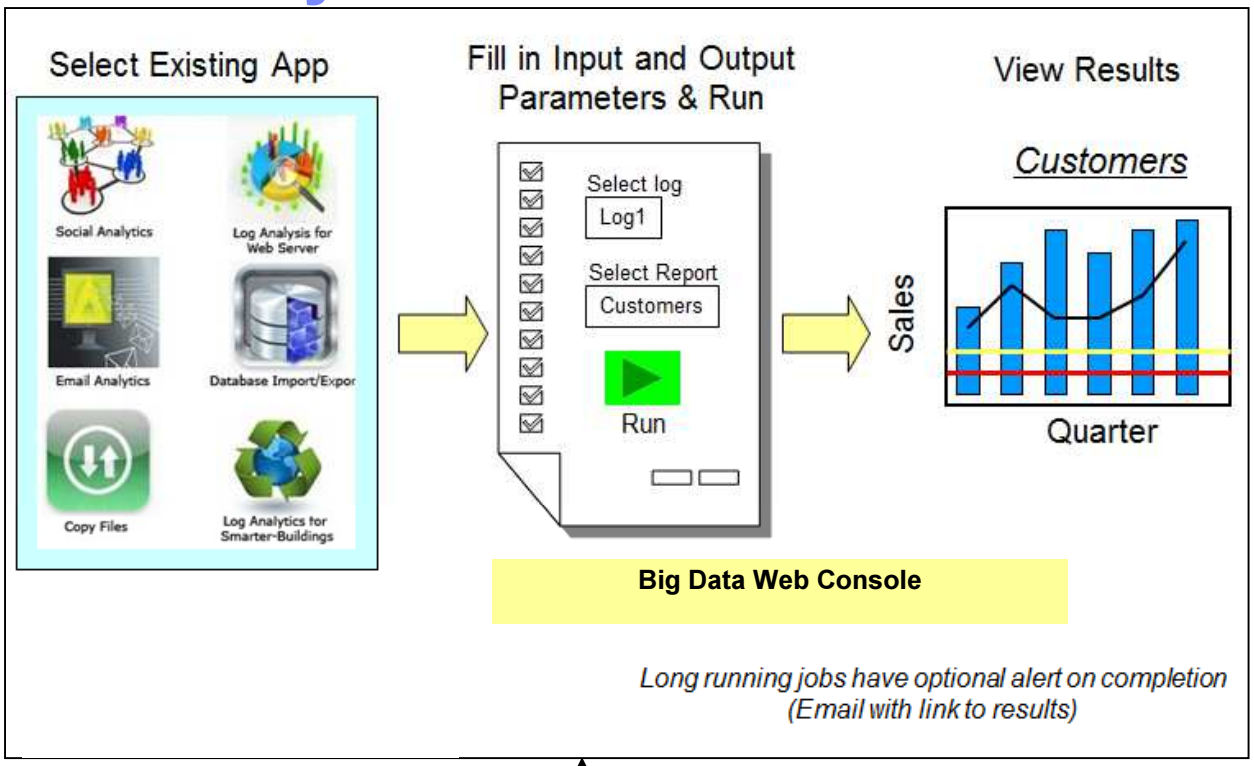
February 8, 2014

Big Data Application Ecosystem

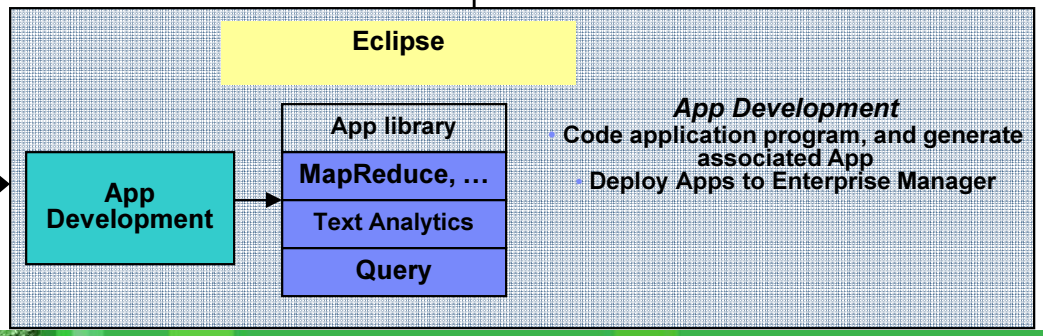
Data integration scenario:
Pre-defined work flows simplify loading data from various sources
- **Work flows can be configured, deployed, executed and scheduled**

Application scenarios (web log, email, social media, …):
- **Samples provide starting point, speed time to value**

Development tooling:
- **Text analytics**
- **MapReduce**
- **Query languages**
- **. . .**

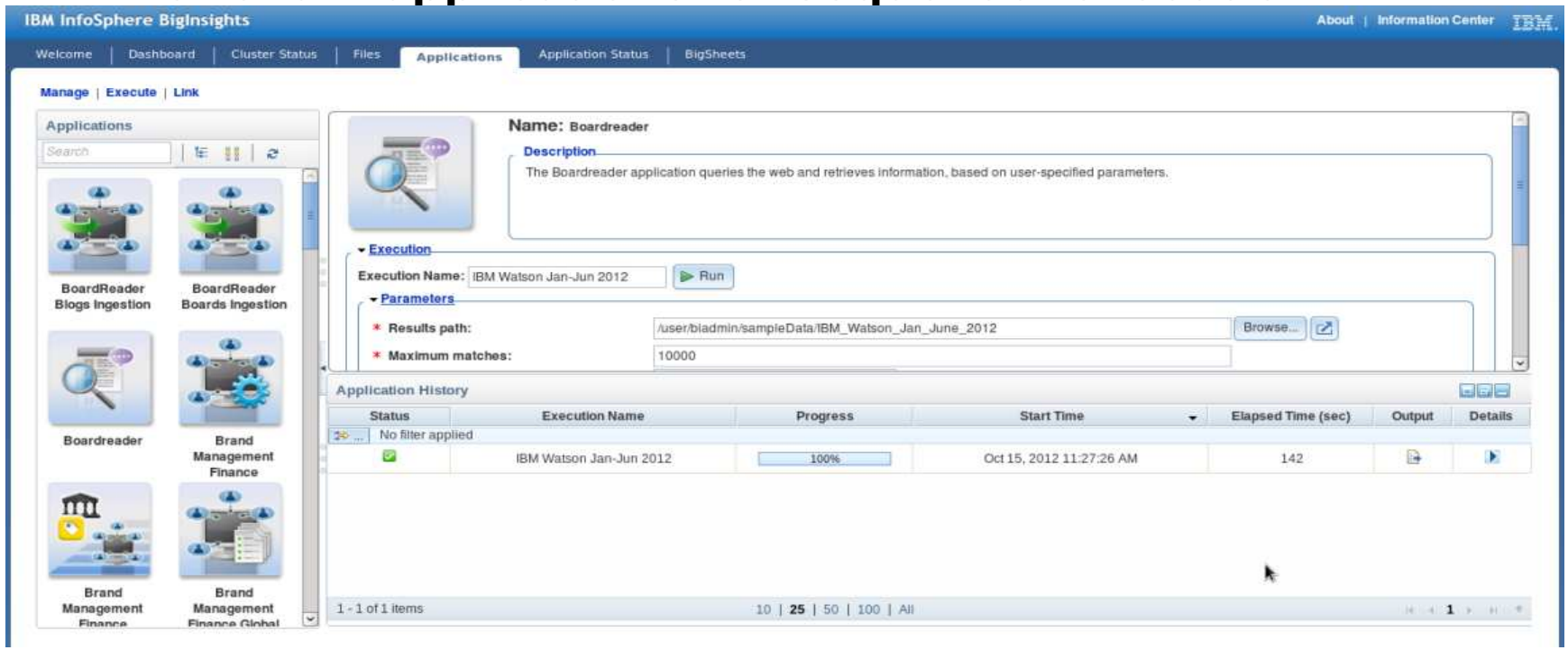


Publish



BigInsights Applications Catalog (Web Console)

- Browse available applications
- Manage and deploy applications (administrators only)
- Execute (or schedule execution of) a deployed application
- Monitor job (application) status
- Link or chain applications for sequential execution

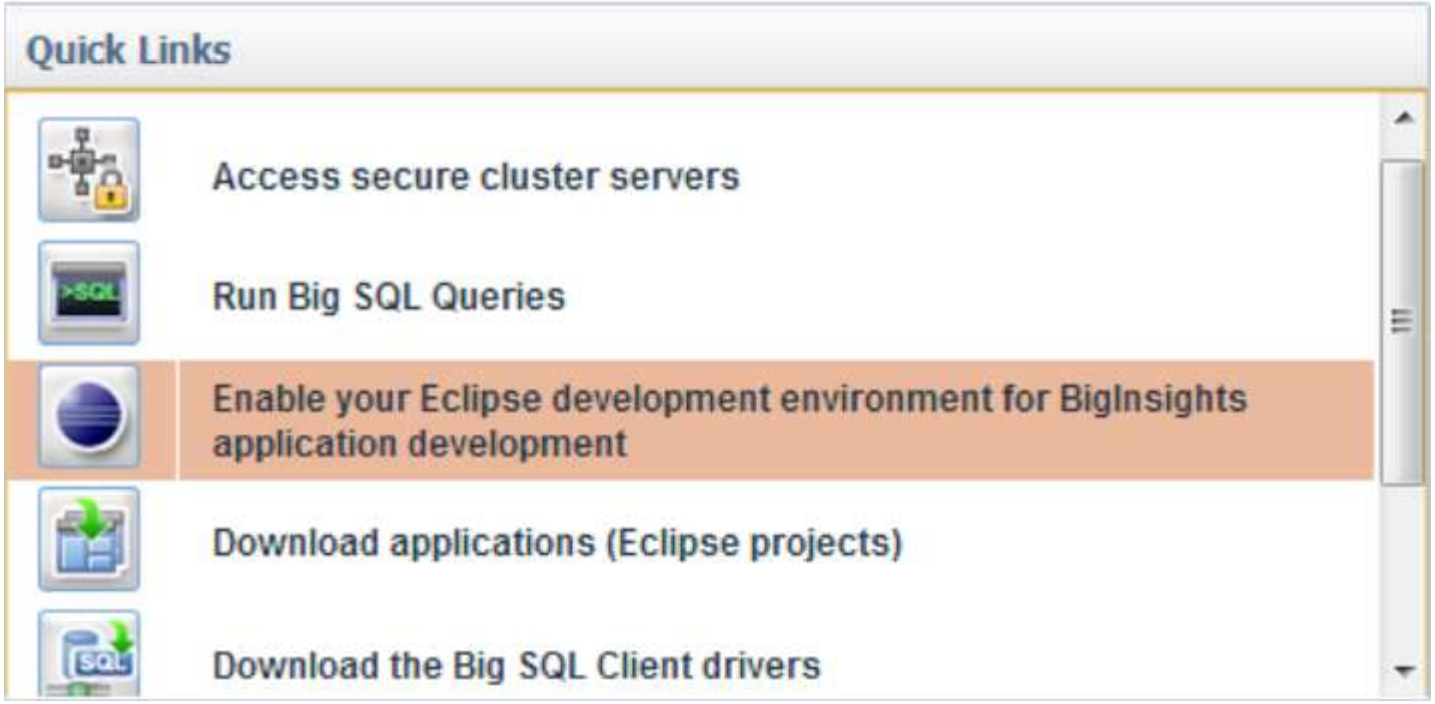


Overview of Application Development Lifecycle

- Configure your Eclipse environment (one-time set up)
- Develop your application using BigInsights tools
- Test your application
- Package and publish your application
- Deploy your application on the cluster

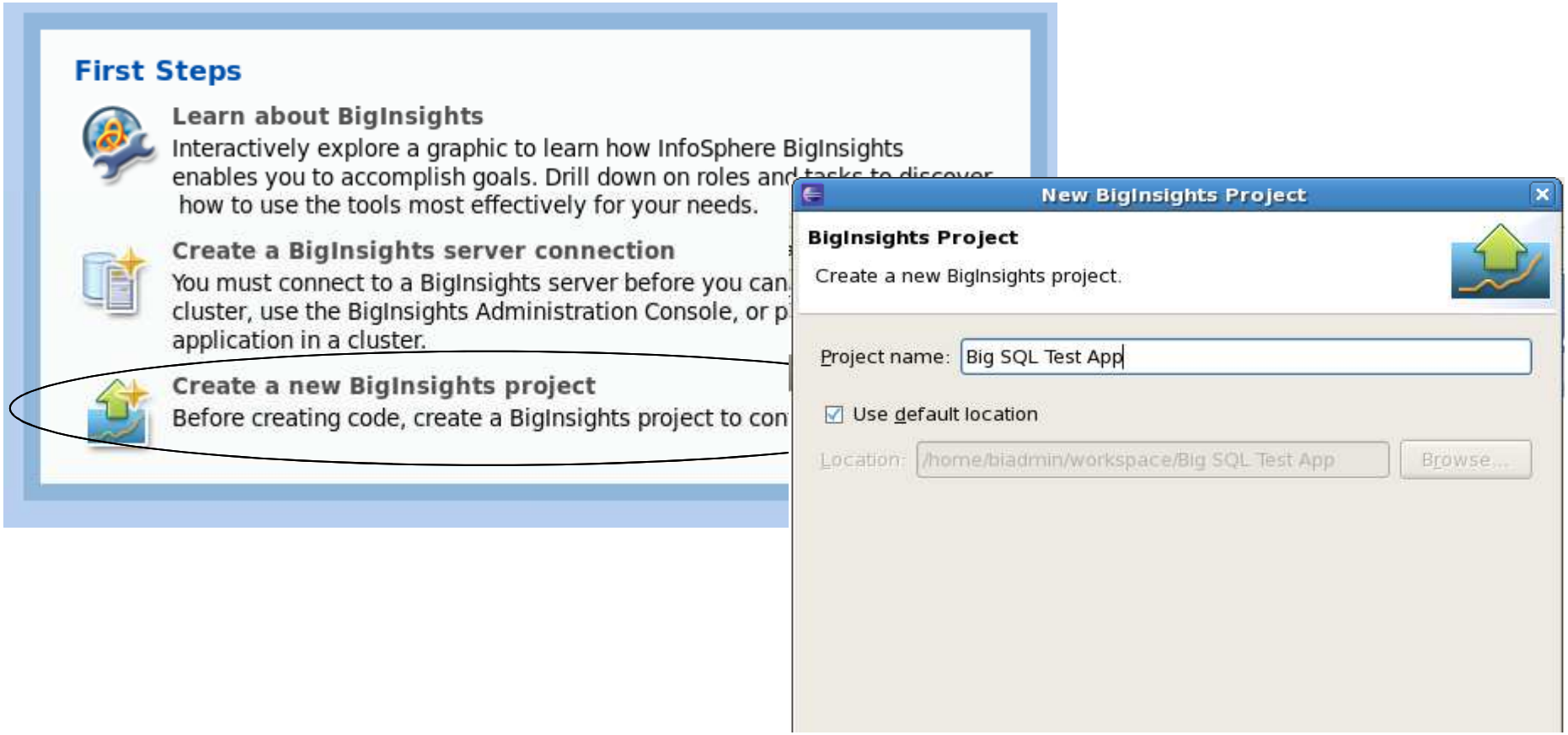
Configure your Eclipse environment

- **One-time set up**
- **Download and install BigInsights tools (Eclipse plug-ins)**
  - Welcome tab of BigInsights Web console includes pre-req info, download & installation instructions



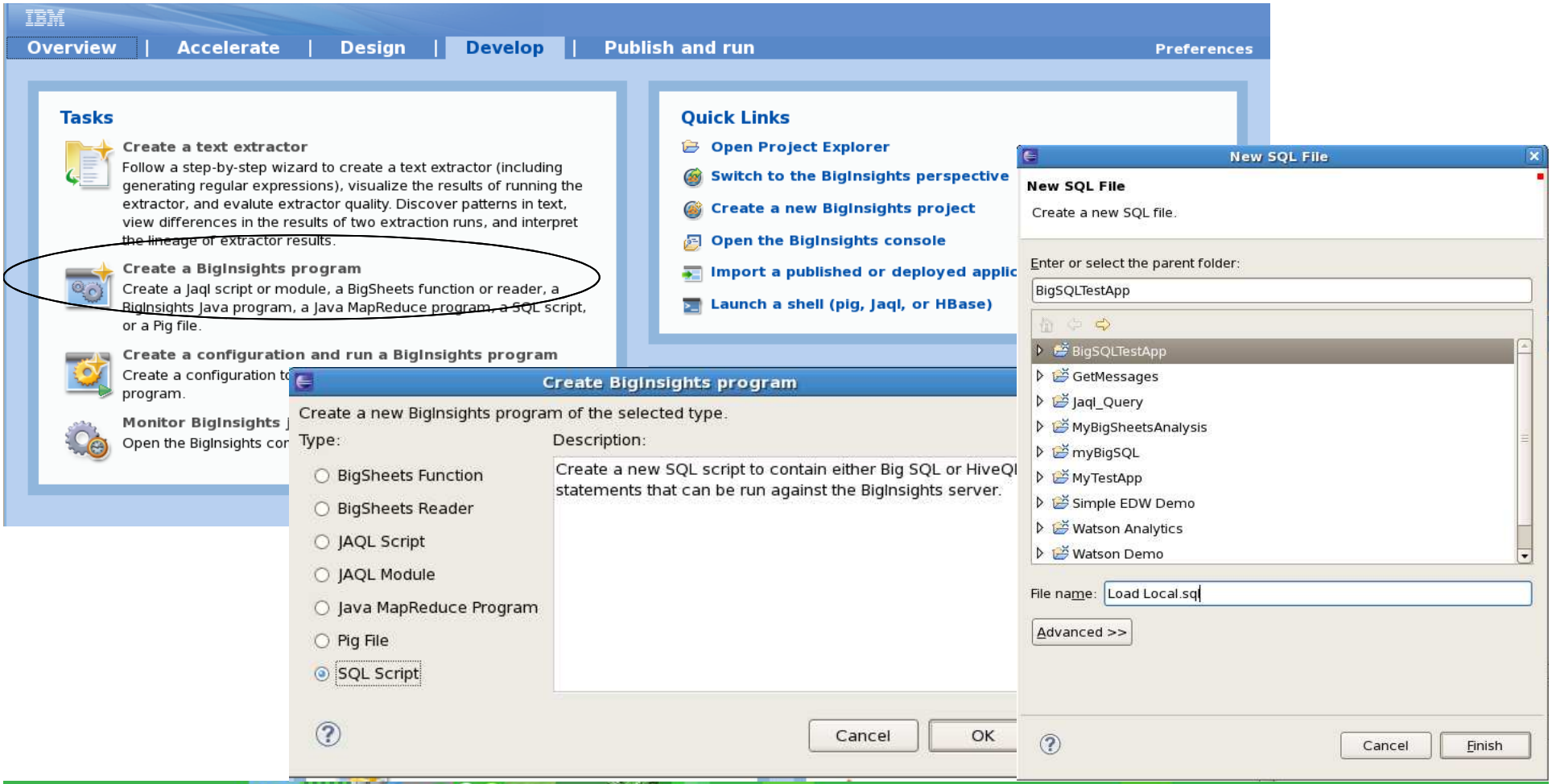
Develop your application – Big SQL example

- Open the BigInsights perspective in Eclipse
- Create a BigInsights project



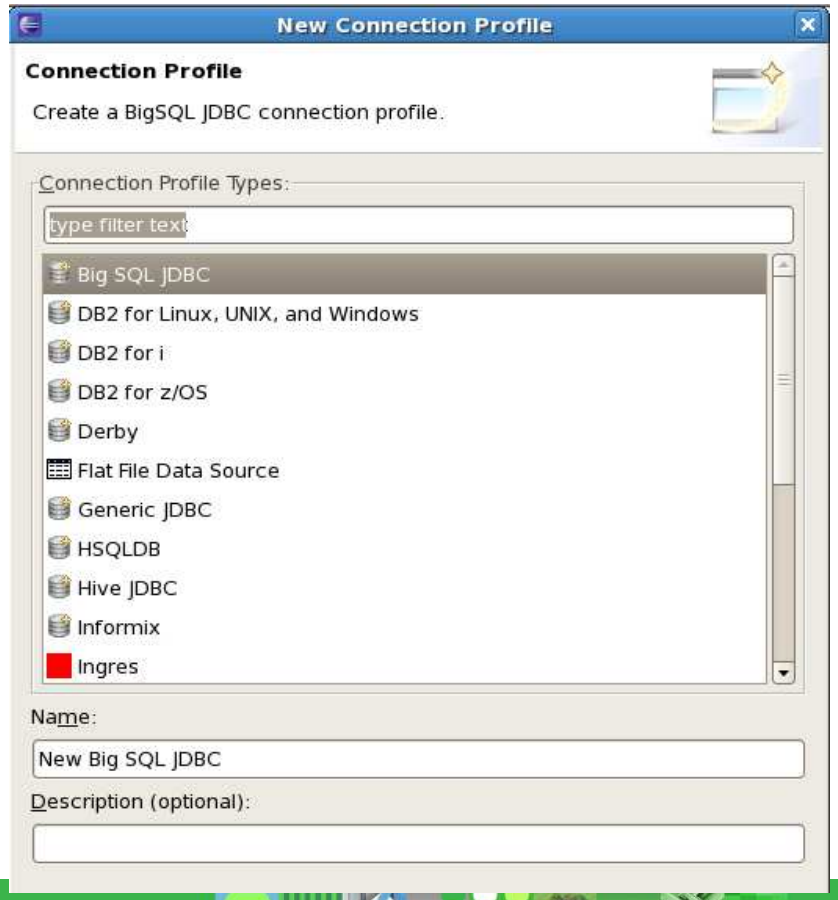
Develop your application – Big SQL example (cont'd)

- **Create a BigInsights program with a SQL script (file)**



Develop your application – Big SQL example (cont'd)

- **Create (or select) a Big SQL connection**



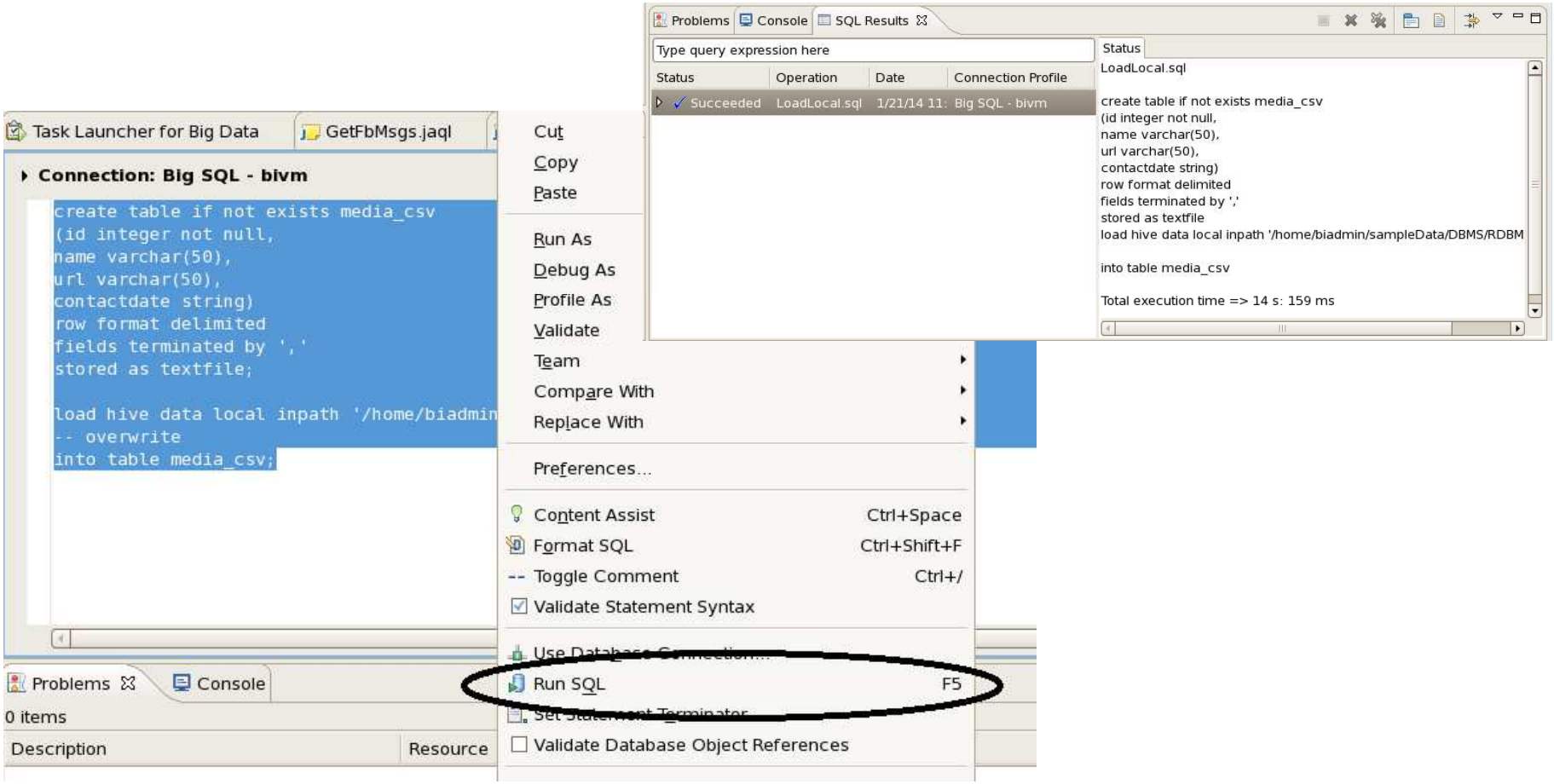
Develop your application – SQL example (cont'd)

- **Populate your SQL file with the desired code**

```
create table if not exists media_csv
(id integer not null,
name varchar(50),
url varchar(50),
contactdate string)
row format delimited
fields terminated by ','
stored as textfile;

load hive data local inpath
'/home/biadmin/sampleData/DBMS/RDBMS_data.csv
'
-- overwrite
into table media_csv;
```

Test your application

- **Run your application from Eclipse**



Publish your application to the BigInsights catalog

- Package and publish your application from Eclipse
- Specify application name, workflow requirements, etc.

BigInsights Application Publish

Specify Application

The information specified here will be saved in the application.xml file.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics

Application

Create New Application

Replace Existing Application

Type a unique application name. Existing names are sh

Name: * BigSQLTestApp

Description: Load data from local file into Big SQL table

Icon: /home/biadmin/sampleData/DBMS/load-icon.jpeg

Preview Icon:

Comma separated list of category names.

Categories: load

< Back | Next > | Cancel | Finish

BigInsights Application Publish

Application Type

Select the type of application to be published.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Select the type of application to publish. The files packaged into the application will depend on the type selected.

Application Type

Workflow

BigSheets

Jaql Module

Text Analytics

BigInsights Application Publish

Specify Workflow

The information specified here will be saved in the workflow.xml file.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Select an existing workflow.xml file.

/home/biadmin/workspace/BigSQLTestApp/BIApp/workflow/workflow.xml | Browse...

Create a new single action workflow.xml file.

Workflow:

Action Type: * Big SQL

Properties:

*	Name	Value	Variable
*	script	LoadLocal.sql	false
	credentials_prop	/user/biadmin/credstore/private/mykeys_Bigl.txt	false

New... | Edit...

BigInsights Application Publish

Zip and Publish Application

Specify the application zip file content. Publish the application to BigInsights.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Zip Preview:

BIApp.zip
- application
  - defaultApp_L.png
  - application.xml
- workflow
  - lib
  - workflow.xml
  - LoadLocal.sql

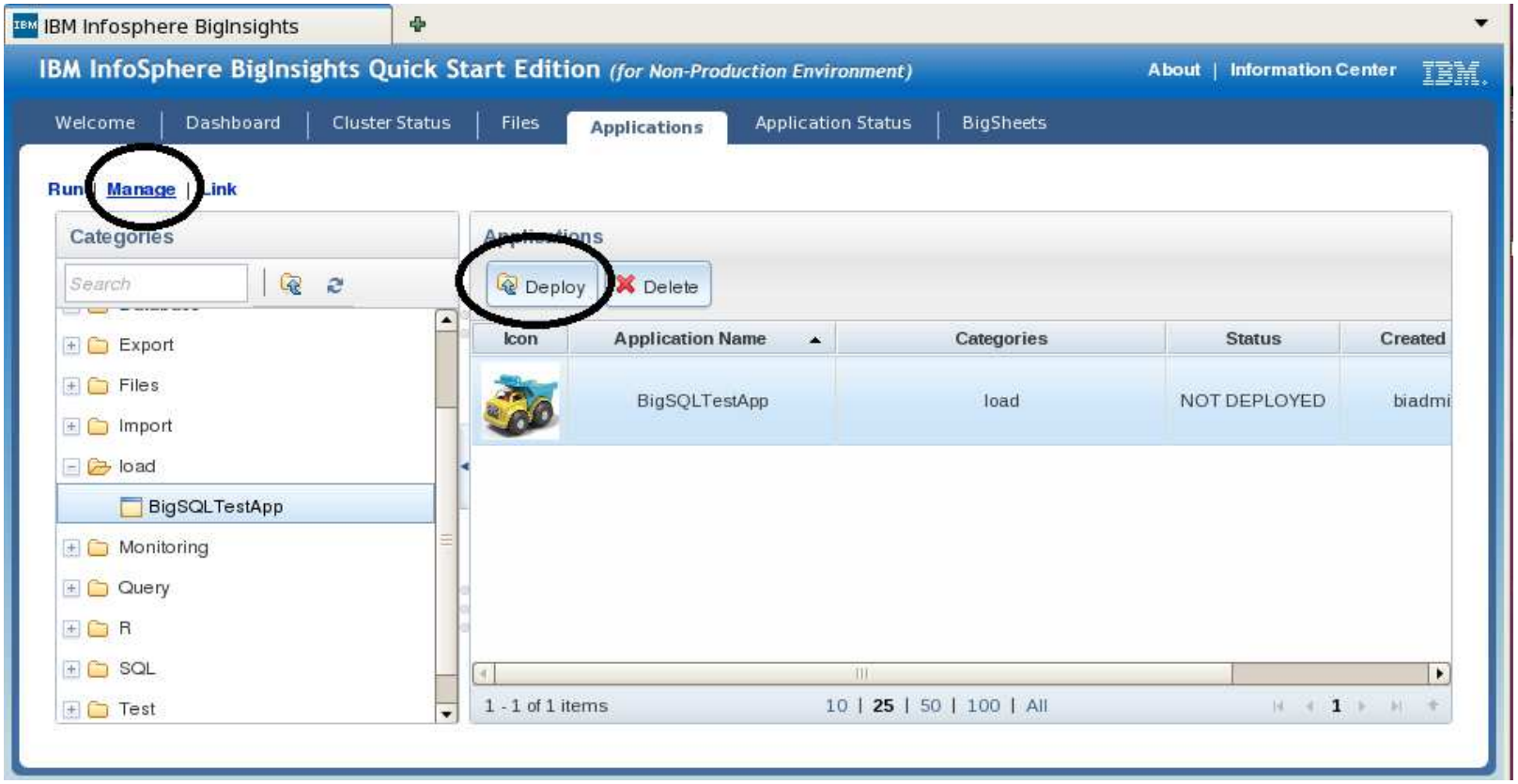
Add... | Remove | Change Path... | Add JAQL Module... | Create Jar...

Source:

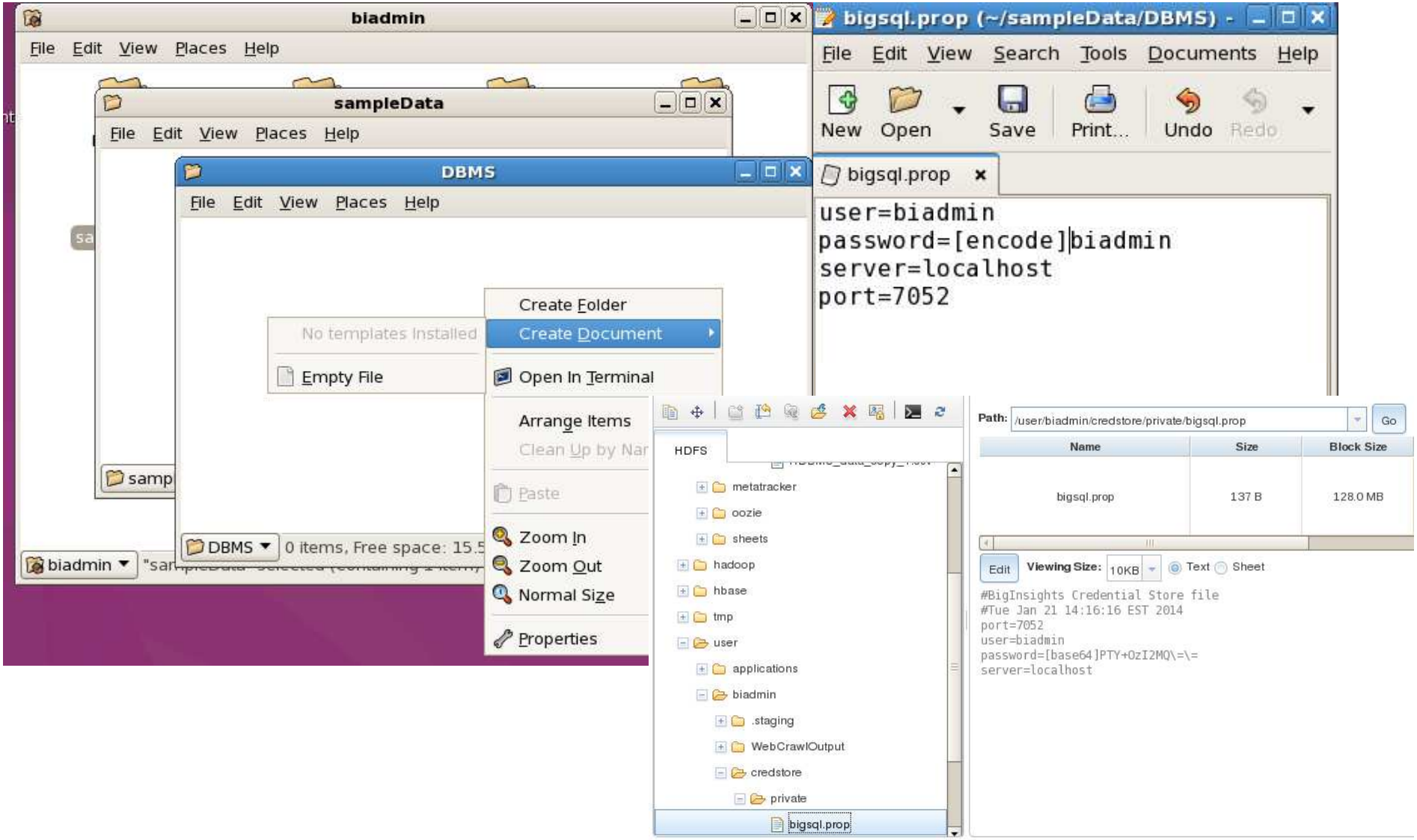
< Back | Next > | Cancel | Finish

Deploy your application on the cluster

- Access the Applications tab of the Web console
- "Manage" the published applications
- Locate your new application and deploy it
- Create credentials store file in DFS (if needed) -- see next chart
- Optionally, execute the application after it's been deployed



Simple credential store properties file



Run your Big SQL application on the cluster

Welcome | Dashboard | Cluster Status | Files | Applications | Application Status | BigSheets

Run | Manage | Link

Applications

Search

BigSQLTestApp

Database Import

Distributed File Copy

BigSQLTestApp

Description

Load data from local file into Big SQL table

Execution

Execution Name: Default

Run

Application History

Status	Execution Name	Progress	Start Time	Elapsed Time (sec)	Details
No filter applied					
	Test	100%	2014 1 21 12:18	58	

1 - 1 of 1 items

Welcome | Dashboard | Cluster Status | Files | Applications | Application Status | BigSheets

HDFS

hdfs://bivm:9000/

biginsights

hive

warehouse

media_csv

RDBMS_data.csv

Path: /biginsights/hive/warehouse/media_csv/RDBMS_data.csv

Name	Size	Block Size
RDBMS_data.csv	606 B	128.0 MB

Edit Viewing Size: 10KB Text Sheet

```
111,"The Business Journals","www.bizjournals.com","2012-01-05-(
222,"CNN","www.cnn.com","2012-01-15-00.00.00"
333,"CIO Today","www.cio-today.com","2012-02-12-00.00.00"
444,"Forbes","www.forbes.com","2012-01-15-00.00.00"
```

Upgrade your application (optional)

- **Satisfy evolving business requirements, improve flexibility**
  - Example: add input parm(s)
- **Modify code and re-package, re-publish, re-deploy**

```
create table if not exists $TABLE
(id integer not null,
name varchar(50),
url varchar(50),
contactdate string)
row format delimited
fields terminated by ','
stored as textfile;

load hive data local inpath
'/home/biadmin/sampleData/DBMS/RDBMS_data.csv'
-- overwrite
into table $TABLE;
```

Upgrade your application (con'td)

- Re-publish your application from Eclipse
- Adjust workflow specs for input parm

BigInsights Application Publish

Specify Application

The information specified here will be saved in the application.xml file.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets

Application

Create New Application

Replace Existing Application

Application to replace

Name: BigSQLTestApp

Description: Big SQL sample application to load data into a table. Uses 1 input p

Icon: /home/biadmin/workspace/BigSQLTestApp/defaultApp_L.png

Preview Icon:

BigInsights Application Publish

Application Type

Select the type of application to be published.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Select the type of application to publish. The files packaged into the application will depend on the type selected.

Application Type

Workflow

BigSheets

Jaql Module

Text Analytics

BigInsights Application Publish

Specify Workflow

The information specified here will be saved in the workflow.xml file.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Select an existing workflow.xml file.

/home/biadmin/workspace/BigSQLTestApp/BIApp/workflow/workflow.xml

Browse...

Create a new single action workflow.xml file.

Workflow:

Action Type: * Big SQL

Properties:

*	Name	Value	Variable
*	script	LoadLocal.sql	false
	credentials_prop	/user/biadmin/credstore/private/mykeys_Bigl.txt	false
	variable	TABLE="${table}"	false

New... Edit... Remove

BigInsights Application Publish

Zip and Publish Application

Specify the application zip file content. Publish the application to BigInsights.

Location ⇨ Application ⇨ Type ⇨ Workflow ⇨ Text Analytics ⇨ BigSheets ⇨ Parameters ⇨ Publish

Zip Preview:

- BIApp.zip
  - application
    - defaultApp_L.png
    - application.xml
  - workflow
    - lib
    - workflow.xml
    - LoadLocal.sql

Add... Remove Change Path... Add JAQL Module... Create Jar...

Source:

< Back Next > Cancel Finish

Summary

- **Eclipse tools simplify big data application development for BigInsights**
  - Wizards
  - Context-sensitive help
  - Oozie workflow generation
  - Built-in test environment
  - Etc.
- **Application catalog provides easy way to locate and launch apps of interest**
  - Developers use Eclipse tools to package/publish their applications to this catalog
  - Application upgrades easily managed