Building Big Data Applications with SQL-MapReduce and Aster Developer Express

Title of Presenter | Date

Peter Pawlowski, Software Engineer
Big Data: The Trend

Many companies have an ever-increasing amount of data.
Big Data: The Challenge

How do we enable all these companies to build useful applications using this data?

“Big Data Applications”
Not Everyone is Google
Our Goal: Big Data Applications Made Easy

1. We enable storing terabytes to petabytes of data in our scalable, parallel database.

2. We provide a framework for running rich, interesting analytics on this data.
Our Goal: Big Data Applications Made Easy

1. We enable storing terabytes to petabytes of data in our scalable, parallel database.

2. We provide a framework for running rich, interesting, and fast analytics on this data.
Reports, Advanced Analytics, Applications (SQL, SQL-MapReduce, ODBC, JDBC)

Aster Data nCluster

Queen

Queries

Workers (Parallel Processing)

Loading/Export

Backup

Data
We started with a SQL interface

• Great fit for many data problems

• Many people know it

• Plugs into many applications & tools
... but SQL ...

- Is a bad fit for many kinds of analytic applications
- Can be hard to write & maintain
- Can be hard to execute efficiently
Enter SQL-MapReduce

• SQL-MapReduce is our framework for pushing analytic code into the database

• Enables much richer analytics to happen on the data in the database
Two Steps to SQL-MapReduce

• Write a SQL-MapReduce function in Java*

• Invoke it from SQL

* Other languages supported, too. See our website.
import java.util.ArrayList;

public final class T implements RowFunction {

    // The constructor establishes the RuntimeContract between
    // your function and nCluster. During query planning, your
    // function will constructed on a single node. During query
    // execution, it will be constructed and run on one or more
    // nodes.
    
    public T(RuntimeContract contract) {
        // Construct the output schema.
        List<ColumnDefinition> outputColumns = new ArrayList<ColumnDefinition>();
        outputColumns.addAll(contract.getInputInfo().getColumns());
    }
}
Aster Data Developer Express

Develop → Test → Deploy

SQL/SQL-MR App → Test Environment → Aster Data nCluster

Confidential and proprietary. Copyright © 2010 Aster Data Systems, Inc.
Step 1: Develop
Step 2: Test Locally

- Run tests on your own machine without requiring an nCluster instance.
- Configure with JSON.
Step 2: Test Locally

- Run tests on your own machine without requiring an nCluster instance.
- Configure with JSON.
Step 2: Test Locally
Step 3: Deploy

Add a new nCluster connection

The connection information was successfully validated.

<table>
<thead>
<tr>
<th>Name:</th>
<th>VMCluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Specified hostname and database to connect:</td>
<td></td>
</tr>
<tr>
<td>Hostname:</td>
<td>10.50.94.100</td>
</tr>
<tr>
<td>Database :</td>
<td>beehive</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Specify the user authentication information:</td>
<td></td>
</tr>
<tr>
<td>Username :</td>
<td>beehive</td>
</tr>
<tr>
<td>Password :</td>
<td>******</td>
</tr>
<tr>
<td>Re-type Password :</td>
<td>******</td>
</tr>
</tbody>
</table>

[Image: screenshot of the interface showing the connection details]
Step 3: Deploy

- Push application into Aster database directly from within IDE
- Initiate and validate application with actual data stored inside $n$Cluster
Aster Data Developer Express is freely available

Download at

www.asterdata.com

Questions?

peter@asterdata.com