Introduction

• Dell is partnering with MySQL to provide best practices white papers such as

• Available at dell.com/mysql

• This presentation explores three methods for building web applications with MySQL back end:
  – PHP, ASP, JSP
Application - Architecture

Application Layer
- Apache +
  - PHP
  - ASP w/Mono
  - JSP

Database Layer
- MySQL Database
  - Seven Tables
  - Transactions
  - Stored Procs
  - 100 GB
The MySQL Database Schema

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
<th>Number of Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMERS</td>
<td>CUSTOMERID, FIRSTNAME, LASTNAME, ADDRESS1, ADDRESS2, CITY, STATE, ZIP, COUNTRY, REGION, EMAIL, PHONE, CREDITCARDTYPE, CREDITCARD, CREDITCARDEXPIRATION, USERNAME, PASSWORD, AGE, INCOME, GENDER, PROD_ID_IDX, PROD_ID1, PROD_ID2 ... PROD_ID10</td>
<td>200 million</td>
</tr>
<tr>
<td>ORDERS</td>
<td>ORDERID, ORDERDATE, CUSTOMERID, NETAMOUNT, TAX, TOTALAMOUNT</td>
<td>120 million</td>
</tr>
<tr>
<td>ORDERLINES</td>
<td>ORDERLINEID, ORDERID, PROD_ID, QUANTITY, ORDERDATE</td>
<td>600 million</td>
</tr>
<tr>
<td>PRODUCTS</td>
<td>PROD_ID, CATEGORY, TITLE, ACTOR, PRICE, SPECIAL, COMMON_PROD_ID1, COMMON_RATING1, COMMON_PROD_ID2, COMMON_RATING2, COMMON_PROD_ID3, COMMON_RATING3</td>
<td>1 million</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>PROD_ID, QUAN_IN_STOCK, SALES</td>
<td>1 million</td>
</tr>
<tr>
<td>REORDER</td>
<td>PROD_ID, DATE_LOW, QUAN_LOW, DATE_REORDERED, QUAN_REORDERED, DATEEXPECTED</td>
<td>variable</td>
</tr>
<tr>
<td>CATEGORIES</td>
<td>CATEGORY, CATEGORYNAME</td>
<td>16</td>
</tr>
</tbody>
</table>
DVD Store

Returning customer? Please enter your username and password
Username: [ ] Password: [ ] Login

New customer? Please click New Customer
New Customer

Thank You for Visiting the DVD Store!

Copyright © 2005 Dell
Application – New Customer Page

DVD Store

New Customer - Please Complete All Required Fields Below (marked with *)

- Firstname: Nicolas
- Lastname: Pujo
- Address1: 1 Easy Street
- City: Austin
- State: Texas
- Zipcode: 78730
- Country: United States
- Email: mbloos@pujo.com
- Phone: 123-456-7890
- Credit Card Type: Dell Preferred
- Credit Card Number: 1234567890123456
- Credit Card Expiration: Jun
- Username: nickyp
- Password: ********
- Age: 22
- Income ($US): $500,000
- Gender: Male

Submit New Customer Data
Application – Browse Page

DVD Store

Select Type of Search

- Title
- Actor [DEPP]
- Category [Documentary]

Number of search results to return [5]
Search

Search Results

<table>
<thead>
<tr>
<th>Add to Shopping Cart</th>
<th>Title</th>
<th>Actor</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AFRICAN WISDOM</td>
<td>ED DEPP</td>
<td>13.99</td>
</tr>
<tr>
<td></td>
<td>ADAPTATION WARS</td>
<td>TIM DEPP</td>
<td>22.99</td>
</tr>
<tr>
<td></td>
<td>ALADDIN ALICE</td>
<td>MAE DEPP</td>
<td>16.99</td>
</tr>
<tr>
<td></td>
<td>ACE JERKLY</td>
<td>JON DEPP</td>
<td>27.99</td>
</tr>
<tr>
<td></td>
<td>ALADDIN WITCHES</td>
<td>ETHAN DEPP</td>
<td>24.99</td>
</tr>
</tbody>
</table>

Update Shopping Cart

Thank You for Visiting the DVD Store!
Application – Purchase Page

DVD Store

Selected Items: specify quantity desired; click Purchase when finished

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Title</th>
<th>Actor</th>
<th>Price</th>
<th>Remove From Order?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>AFRICAN WISDOM</td>
<td>ED DEPP</td>
<td>13.99</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>ADAPTATION WARS</td>
<td>TIM DEPP</td>
<td>22.99</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>ACE JEKYL</td>
<td>JON DEPP</td>
<td>27.99</td>
<td></td>
</tr>
</tbody>
</table>

Subtotal: 120.94
Tax (8.25%): 9.93
Total: 130.92

Thank You for Visiting the DVD Store!

Copyright © 2005 Dell
Application Mid-tier Implementations

- PHP
- ASP
- JSP
PHP - Introduction

• PHP is a widely-used general-purpose scripting language that is especially suited for Web development
• Excellent integration with MySQL
• See http://www.php.net
PHP – Components/Setup

• RPM: php-mysql-4.3.2-8.ent.i386.rpm
• Edit /usr/share/mysql/my-large.cnf, copy to /etc/my.cnf
  – add following line to mysqld section:
  – log=/var/lib/mysql/mysql_query.log
• modify /etc/php.ini:
  – mysql.default_host = 192.168.224.11
  – mysql.default_user = web
if (!($link_id=mysql_connect())) die("Can't connect");
$query = "select COUNT(*) from DS2.CUSTOMERS where USERNAME='$username';";
$result = mysql_db_query('DS2',$query);
$check_username = mysql_fetch_row($result);
if ($check_username[0] != 0)
{
    ds_html_header("New Customer Login");
    echo "<H2>Username already in use! Please try another username</H2>\n";
    ...
}
else
{
    $insert_newuser_query = "INSERT INTO DS2.CUSTOMERS (FIRSTNAME, LASTNAME... VALUES ('$firstname','$lastname,...
$insert_newuser_result = mysql_db_query('DS2',$insert_newuser_query);
$customerid = mysql_insert_id($link_id);
PHP – MySQL Transactions

$command_result = mysql_query('START TRANSACTION;');
$purchase_insertorder_query = "INSERT into DS2.ORDERS (ORDERDATE, ...
  VALUES ($currentdate ...
$purchase_insertorder_result = mysql_query($purchase_insertorder_query);
$orderid = mysql_insert_id();
...
$query = "SELECT QUAN_IN_STOCK, SALES FROM DS2.INVENTORY WHERE PROD_ID=$item[$h]";
...
$new_quan = $curr_quan - $quan[$h];
if ($new_quan < 0)
  {
    echo "Insufficient quantity for prod $item[$h]\n";
    $success = false;
  }
else
  {
    $query = "UPDATE DS2.INVENTORY SET QUAN_IN_STOCK=$new_quan, SALES=$new_sales WHERE
      PROD_ID=$item[$h];";
    $result = mysql_query($query);
  }
...
if ($success) mysql_query('COMMIT;');
else mysql_query('ROLLBACK;');
PHP5 – Stored Procedure Call

$new_customer_proc_call = "call DS2.NEW_CUSTOMER(" . 
""$firstname','$lastname','$address1','$address2','$city','$state','$zip','$country'," . 
""$region','$email','$phone','$creditcardtype', '$creditcard','$creditcardexpiration'," . 
""$username','$password','$age','$income','$gender',@customerid_out);"; 

mysqli_query($link_id, $new_customer_proc_call);

$query = "select @customerid_out;";
mysqli_real_query($link_id, $query);
$result = mysqli_use_result($link_id);
$row = mysqli_fetch_row($result);
$customerid = $row[0];
mysqli_free_result($result);
ASP - Introduction

- ASP.NET – Microsoft Active Server Pages for .NET
- Can program in VB.NET or C#
- Runs under Windows with MS IIS (Internet Information Server) or under Linux/Apache with Mono
- Mono: open source Common Language Runtime environment from Ximian
  – See mono (http://www.mono-project.com)
ASP – MySQL Connector

conn_str = "Server=w2k31;User ID=root;Password=password;Database=DS2;"
conn = new MySqlConnection(conn_str);
conn.Open();
db_query="select count(*) FROM DS2.CUSTOMERS where
    USERNAME='"+username+"';";
cmd = new MySqlCommand(db_query, conn);
if(cmd.ExecuteScalar() != null) count =
    Convert.ToInt32(cmd.ExecuteScalar().ToString());
if (count > 0) // Non-unique username
{
    Response.Write("<H2>Username already in use! Please try another
        username</H2>\n")
    ...
}
else // Unique username - insert customer data into CUSTOMERS table
{
    ...
    db_query =
        "INSERT INTO DS2.CUSTOMERS (FIRSTNAME,LASTNAME,...
        " VALUES ('" + firstname + "','" + lastname + ",',"
    cmd = new MySqlCommand(db_query, conn);
    cmd.ExecuteNonQuery();
db_query = "select LAST_INSERT_ID();";
cmd = new MySqlCommand(db_query, conn);
if(cmd.ExecuteScalar() != null) customerid =
    Convert.ToInt32(cmd.ExecuteScalar().ToString());
trans = conn.BeginTransaction(IsolationLevel.RepeatableRead); // default is ReadCommitted

db_query = "INSERT into DS2.ORDERS (ORDERDATE, CUSTOMERID, ...
  "VALUES('' + currentdate + '','' + customerid + '''');
cmd = new MySqlCommand(db_query, conn, trans);
cmd.ExecuteNonQuery();

db_query = "select LAST_INSERT_ID();";
cmd = new MySqlCommand(db_query, conn);
if (cmd.ExecuteScalar() != null)
  orderid = Convert.ToInt32(cmd.ExecuteScalar().ToString());
if (orderid > 0) success = true;
...

db_query = "SELECT QUAN_IN_STOCK, SALES FROM DS2.INVENTORY WHERE " +
  "PROD_ID=" + item[i] + ";";
cmd = new MySqlCommand(db_query, conn);
reader = cmd.ExecuteReader();
reader.Read();
curr_quan = reader.GetInt32(0);
new_quan = curr_quan - quan[i];
if (new_quan < 0) {
  ...
  success = false;
}
else {
  
  db_query = "UPDATE DS2.INVENTORY SET QUAN_IN_STOCK=" + new_quan + ", SALES=" +
    new_sales + " WHERE PROD_ID=" + item[i] + ";"
  cmd = new MySqlCommand(db_query, conn, trans);
  cmd.ExecuteNonQuery();
}
if (success) trans.Commit();
else trans.Rollback();
JSP – Introduction

• JSP – Java Server Pages
• Allows for Java code to be put directly into web page
• Within an HTML page java code is identified by <%= java code here %>.
• Tomcat is the reference implementation for JSP
  – Part of Apache subproject Jakarta
  – Part of JBoss
JSP-Components/Setup

• Components
  – J2SE5 from java.sun.com for linux
  – TomCat 5.5.7 from jakarta.apache.org
  – MySQL J Connector 3.1.7 from www.mysql.com

• Setup
  – Extract java and tomcat to desired locations
  – Include JConnector in classpath for TomCat
  – export CATALINA_HOME
  – export JRE_HOME
  – $CATALINA_HOME/bin/startup.sh
  – http://<hostname>:8080
JSP - MySQL Connection

- Define the mysql driver from J Connector to be used
  - J Connector implements JDBC API for MySQL
- Create connection object, then create statement object
- Execute query with statement object
- Get query results in a ResultSet object
  - When connection or statement is closed, associated ResultSets are closed
  - next() method for ResultSet object must be called before getting data from resultset object

```java
try { Class.forName("com.mysql.jdbc.Driver"); } 
catch (Exception e) {System.out.println("Error");}
Connection conn = DriverManager.getConnection("jdbc:mysql:///DS2?user=web");
String query = "select CUSTOMERID FROM DS2.CUSTOMERS where USERNAME=" + user_name + " and PASSWORD=" + password + ";";
Statement userqueryStatement = conn.createStatement();
ResultSet userqueryResult = userqueryStatement.executeQuery(query);
if (userqueryResult != null && userqueryResult.next())
```
JSP - MySQL Insert

- Create Connection to MySQL and associated Statement object
- Execute Statement with executeUpdate because of INSERT
  - Statement.RETURN_GENERATED_KEYS allows for the auto increment Customer Id column to be returned
- Use getGeneratedKeys() method of Statement object to retrieve the customer id of newly inserted customer
  - getGeneratedKeys() is part of JDBC spec and is equivalent to LAST_INSERT_ID() mysql function

try {
    Class.forName("com.mysql.jdbc.Driver");
} catch (Exception e) {System.out.println("Error opening connection");}
Connection newuserconn = DriverManager.getConnection("jdbc:mysql://localhost/DS2?user=web");
Statement userInsertStatement = newuserconn.createStatement();
userInsertStatement.executeUpdate(insert_newuser_query,Statement.RETURN_GENERATED_KEYS);
    // the RETURN_GENERATED_KEYS option on the executeUpdate is needed for the
    // autoincrement customerid column to be returned.
ResultSet userInsertResult = userInsertStatement.getGeneratedKeys();
    // Get the auto generated key into a result set
userInsertResult.next();
String customerid = userInsertResult.getString(1); // get autogenerated customerid into string
JSP - MySQL Transaction

- Create a connection and change default for autocommit to false, a commit is issued at the end of transaction
- If any SQL errors occur – then a rollback is issued in the catch

```java
try {
    orderconn = DriverManager.getConnection("jdbc:mysql://DS2?user=web");
    orderconn.setAutoCommit(false); // tell connection to not commit until instructed
    Statement purchaseupdateStatement = orderconn.createStatement();
    purchaseupdateStatement.executeUpdate(purchase_insert_query, Statement.RETURN_GENERATED_KEYS);
    ResultSet orderIDResult = purchaseupdateStatement.getGeneratedKeys(); // to get orderid that is autogenerated by db
    orderid = orderIDResult.getString(1);
    // loop through purchased items and make inserts into orderdetails table
    ...
    purchaseupdateStatement.executeUpdate(p_query); // Insert into orderlines
    purchaseupdateStatement.executeUpdate(c_update); // Update customers with recent purchases
    if ( success == true ) // if no errors were found, commit all
        {orderconn.commit();}
    else
        {orderconn.rollback();} // otherwise, rollback
    orderconn.close();
} //end of try for order entry transaction
catch (SQLException e) // if any SQL exceptions were thrown, rollback
{
    if (orderconn != null)
    {
        try { orderconn.rollback(); } catch (SQLException rbexception) {rbexception.getMessage()}
    }
} //end of catch
```
MySQL DVD Store Performance – InnoDB vs. MyISAM

• Initially all tables were InnoDB to support transactions
• Search queries based on title or actor took over 3 seconds
• Solution – Use MyISAM FullText indexing on Products
  – Create InnoDB based Inventory table to handle transactions
  – Make Products MyISAM with FULLTEXT index on Title and Actor columns
  – Search queries reduced to less than half a second
MySQL DVD Store Performance – Tuning Parameters

- Added `innodb_flush_method=O_DIRECT`
  - 4% improvement
- `innodb_thread_concurrency` increased to 20
  - 1% improvement
- Altered memory settings to allow for larger InnoDB Buffer Cache
  - 1% improvement
  - Reduced `key_buffer` to match size of products table
MySQL DVD Store Performance – Stress Test Results

• Dell PowerEdge 2800
  – 2 x 3.0 GHz Intel Xeon
  – 8 Internal 10k RPM SCSI Disks
  – 4 GB RAM
  – Under $9000 for Hardware and Software¹
  – 1900+ Orders Per Minute
  – Complete details at http://www.dell.com/mysql

¹ The Dell PowerEdge 2800 was priced at $8187 with a configuration of 2x3.0 GHz Xeon processors, 4GB RAM, 8 146GB 10k rpm hard disks, 3 year bronze service and support and SuSE Linux Enterprise Server operating system in the small and medium business section of www.dell.com on March 2, 2005. An additional cost of $595 for MySQL Network was added to include the cost of the database software used. Total hardware and software cost was $8782. See "MySQL Network and the Dell PowerEdge 2800: Capacity Sizing and Performance Tuning Guide for Transactional Applications" by Todd Muirhead, Dave Jaffe and Nicolas Pujol (April 2005)(http://www.dell.com/mysql).
Conclusions

• All three languages have their strengths
  – PHP is easiest to interface to MySQL and to code web pages
  – JSP utilizes large population of Java developers
  – ASP.NET/Mono utilizes C# skills, enables use of .NET Framework