Can I Migrate My Database to MySQL and What Will it Cost?

Brian Miezejewski
Practice Manager

Copyright Oracle 2010
Overview
Overview

• Purpose is to:
  • Estimate the effort to do a migration to MySQL
  • Identify and mitigate the risks as early as possible
    • Can’t support the requirements
      • Performance: Size, TPS, etc.
    • Takes long to migrate than expected
• Based on experience delivering
  • Actual Migrations to MySQL
    • SQL Server, Oracle, Sybase, Postgres, etc.
  • MySQL DB Migration Assessment Package
• Not limited to one-off migrations
  • Have applied this to Companies with hundreds of DBs
  • Still useful if you are only migrating only one database
Questions to be Answered by Migration Assessment

• Can I migrate this DB to MySQL?
• What is the destination MySQL Architecture?
  • HA: Replication, DRDB, Cluster?
  • Scaling
  • Engine selection
• What is the impact on the database infrastructure?
• How much effort will it take to do the migrations?
• What are the Risks?
Risks

- Database can’t perform
  - TPS
  - Data Size
  - Connections
- Migration takes to long
  - Under estimate the effort
- Dead-end feature
  - Doesn’t support XYZ
- Hidden issues
  - Problem only shows up in Production
- Donkey Dev ...
• This is the way it works in XYZ
• I have always done it that way
• It will not work!
• It's not a real database if I can’t ...
• No!

• Shows up as poor performance
• MySQL is different, often you need to think a different way to use it effectively
The Steps

1) Identify the Migration Candidate Database Systems

2) Capture Key Metrics for
   a) Database
   b) Application(s)
   c) Requirements
   d) Delivery

3) Review the database support infrastructure

4) Eliminate Systems that can not or should not be migrated

5) Deep Dive for more detail as needed

6) Refine Migration Factors

7) Create effort estimates by combining the Migration Factors with the Migration Metrics
Step: 1 Identify the Migration Candidate Database Systems

• Good Candidates
  • Most any MS SQL Server DB
  • Sybase servers not using the parallel option
  • Postgres, Ingres, and most other open source DBs
  • Database that runs in 8 cores or less (16 core or less with 5.5) on a single server
  • Web facing

• Bad Candidates
  • Vendor application does not support MySQL
  • Oracle RAC system that runs active-active for performance purposes
  • Large (4TB+) Teradata, DB2, Oracle, Informix, Netezza, Data Warehouse solutions
  • Needs more than 8 core (16 cores for MySQL 5.5)
Metric types

- Measures used to estimate effort
  - Number of tables
  - Stored procedures
  - Lines of Code (LOC) per stored procedure
- Architectural design
  - HA Req., current and future Size and TPS
- Direct effort metrics
  - Testing time
  - Production change time
- Measure used to determine feasibility (Risk)
  - Connections, DB Size, TPS, etc.
Step:2 Capture Key Metrics For:
a) Databases

- Server
  - Connections, TPS, Data Size, Hardware, OS
- Features used
  - Constraints, Subqueries, Complex joins
- Table Based metrics
  - Number of tables, columns per table, indexes per table, etc.
  - Foreign Keys
- Stored Procedures and Triggers
  - Parameters per SProc
  - Line of Code (LOC)
Step 2: Capture Key Metrics For:
b) Applications

- Application migration can be the largest cost of a database migration
  - Not so much with: Java, PHP, etc.
  - Almost always: C, C++, etc.

- List of Applications per DB System
  - Database “Touch Points”: How many places in the code access the database?
  - Language: C, PHP, Java, Python, etc.
  - Abstraction layer: Hibernate, etc.
  - Stored Procedure usage

- Other Important Items
  - Architecture: 2 tier, # conns per server, etc.
  - Type of Application: OLTP, Web, Reporting, etc.
Step:2 Capture Key Metrics For:
c) Requirements

- **Availability**
  - What is the required up time?
  - Is read more important than write uptime?

- **Scalability**
  - How will the system grow in the future
    - TPS
    - Data Size
    - New Functionality

- **Maintainability**
  - Maintenance window for upgrades?
  - Understand how Application upgrades will be done in MySQL.
Step: 2 Capture Key Metrics For: d) Delivery

- Testing Effort
  - Data (Can be estimated based on DB Metrics)
  - Developer
  - QA
  - User Acceptance/Staging

- Cut Over
  - What is the window for live data migration
  - Can the old and new system be run in parallel
  - Do the applications need to be modified to support the cut-over, i.e. run on both databases simultaneously?
Step: 3 Review the database support infrastructure

• Monitoring tools
  • What tools are used now? Do they support MySQL?
    • Installation and configuration
    • Training
• Backup Tools
  • Scripts, etc.
• Maintenance Processes
  • Documentation
  • Testing
• DBA, User, Support Staff Training
Step: 4  Eliminate Systems that can not or should not be migrated

- Application is too hard to migrate
  - Uses custom API
  - Proprietary query syntax
- Uses custom middle tier
- Un-supported feature set requirement
  - Parallel query/scan requirements
- Politically sensitive
Step 5: Deep Dive for more detail

- Select a “representative” subset of objects to migrate and do the actual migration.
  - Try to get one of each of the most complex objects
- ~1-2 Days effort per DB System
- Normally driven by the application or stored procedures.
  - Usually 4-7 application touch points or stored procedures
  - Migrate the tables and a small set of data
- Goal is improve the accuracy of the migration “factors”
  - i.e. How long it take to migrate an application touch point
Step 6: Refine Migration Factors

- Each Migration Factor is the effort (time) it takes to migrate some object, for example:
  - 15 minutes per stored procedure (or per 40 LOC)
  - 10 minutes per table
  - 30 minutes per C++ touch-point
- Use results of the Deep dive
- Historical knowledge of developers
- Should include unit testing effort
- Still need some effort for testing and manual review even if you use an automated tool
**Step 7: Create effort estimates by combining the Migration Factors with the Migration Metrics**

- Create a spreadsheet that sums the metrics multiplied by their factors
- Example below is very scaled down

<table>
<thead>
<tr>
<th>Database System</th>
<th>Stored Procedures</th>
<th>Effort per Stored Procedure</th>
<th>Total for all Stored Procedures</th>
<th>Application Touch Points</th>
<th>Effort Per Touch Point</th>
<th>Total for All Touch Points</th>
<th>Testing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANDFISH</td>
<td>17</td>
<td>30</td>
<td>510</td>
<td>2</td>
<td>15</td>
<td>45</td>
<td>675</td>
<td>2</td>
</tr>
<tr>
<td>BACKDOOR</td>
<td>75</td>
<td>30</td>
<td>2250</td>
<td>5</td>
<td>34</td>
<td>30</td>
<td>1020</td>
<td>3</td>
</tr>
<tr>
<td>DOGRUN</td>
<td>23</td>
<td>10</td>
<td>230</td>
<td>1</td>
<td>23</td>
<td>10</td>
<td>230</td>
<td>1</td>
</tr>
<tr>
<td>SERVER07</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>102</td>
<td>4590</td>
<td>10</td>
</tr>
<tr>
<td>IMTTRA</td>
<td>52</td>
<td>30</td>
<td>1560</td>
<td>4</td>
<td>47</td>
<td>10</td>
<td>470</td>
<td>1</td>
</tr>
<tr>
<td>JOES</td>
<td>47</td>
<td>30</td>
<td>1410</td>
<td>3</td>
<td>72</td>
<td>30</td>
<td>2160</td>
<td>5</td>
</tr>
<tr>
<td>BDT03</td>
<td>15</td>
<td>90</td>
<td>1350</td>
<td>3</td>
<td>98</td>
<td>45</td>
<td>4410</td>
<td>10</td>
</tr>
</tbody>
</table>
Step 7: Create effort estimates by combining the Migration Factors with the Migration Metrics (Using Tool)

- Assume that the tool will Migrate 80% of the Database objects

<table>
<thead>
<tr>
<th>Database System</th>
<th>Stored Procedures</th>
<th>Effort per Stored Procedure</th>
<th>Total for all Stored Procedures</th>
<th>Application Touch Points</th>
<th>Effort Per Touch Point</th>
<th>Total for All Touch Points</th>
<th>Testing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Manual</td>
<td>Minutes</td>
<td></td>
<td>Minutes</td>
<td></td>
<td>Days</td>
<td>Days</td>
</tr>
<tr>
<td>SANDFISH</td>
<td>17</td>
<td>3.4</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>BACKDOOR</td>
<td>75</td>
<td>15</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>34</td>
<td>10</td>
</tr>
<tr>
<td>DOGRUN</td>
<td>23</td>
<td>4.6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>SERVER07</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>102</td>
<td>14</td>
</tr>
<tr>
<td>IMTTRA</td>
<td>52</td>
<td>10.4</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>47</td>
<td>1</td>
</tr>
<tr>
<td>JOES</td>
<td>47</td>
<td>9.4</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>72</td>
<td>11</td>
</tr>
<tr>
<td>BDT03</td>
<td>15</td>
<td>3</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td>98</td>
<td>17</td>
</tr>
</tbody>
</table>

Tuesday, April 13, 2010
Risk Mitigation

- Donkey Dev
  - Training
  - Experience with MySQL
- Can’t Perform
  - Review performance metrics
- Bad effort estimations
  - Good Metrics
  - Deep Dive
- Dead-end Features
  - Deep Dive
- Hidden Issues
  - Deep Dive + Prototype
Tools

- SQLWays - Java based migration tool
  - Can migrate 80%+ of most stored procedures
  - Everything but the applications, and even a little of that

- MySQL Migration Tool
  - Data and Tables

- Scripts
  - JAVA
  - Perl
  - etc.
The presentation is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.