Minimizing Testing Overheads in Database Migration Lifecycle

Sangameshwar Patil, Sasanka Roy, John Augustine, Amanda Redlich, Sachin Lodha, Harrick Vin, Anand Deshpande, Mangesh Gharote & Ankit Mehrotra

Systems Research Lab (SRL)
Motivation: Need for DB Migration

- Functional enhancements
- Expiry of vendor support for older versions
- DB and server consolidation
  - reduce Total cost of ownership
- Mergers and acquisitions
  - reconciliation of hardware/software platforms
- Performance enhancements, workload balancing

Periodic and daunting task:
Migrate a set of production databases from one HW/SW platform to another
DB Migration Process

• Basic setup and code updates
  - Identify dependent applications
  - Code updates / conflicting SQL
  - Setup, configuration of target/test environment

• ETL
  - Schema change implementation, if any
  - Data cleansing (optional)
  - Back up the source DB

• DB configuration
  - Index generation, user roles, access rights

• Testing and Verification
Constraints on Migration Process

• **Minimize business impact**
  - Migrate only on non-business days

• **Migration cannot happen during blackout-brownouts**

• **Availability**
  - DBAs, Application testing team
  - Target / test hardware platforms
  - Application release schedules

• **Result:** Limit on size/number of databases that can be migrated during a weekend
Migration Cost

• Cost components
  - C1: Basic setup and code-update
  - C2: Migration of individual DB (ETL, configuration)
  - C3: Testing all dependent applications
• C1 and C2 are more predictable / well-defined
  - Migration of individual database is well-understood
  - Automated tools exist for migration of individual database
    • QuickMig - CIKM '07, SQLways - Inspirer, SwissSQL
Testing Cost is Susceptible to Variation

• If number of DBs to be migrated is large
  - Migration over multiple weekends (migration waves)

• C3 is susceptible to large variation
  - Many-to-many App-DB dependencies
  - Complex constraints
  - Partitioning of DBs into migration waves is based on intuition and experience of DBAs
  - Little/no formal, quantitative analysis to support decision-making
  - C3 can increase due to re-testing of applications
A Simple Example

Applications

Databases

\[
\begin{align*}
\text{c}_1 &= \{d_1\} \\
\text{c}_2 &= \{d_2\} \\
\text{c}_3 &= \{d_3\} \\
\text{c}_4 &= \{d_4, d_5, d_7\} \\
\text{c}_5 &= \{d_6\} \\
\text{c}_6 &= \{d_8\}
\end{align*}
\]
DBMP as an optimization problem

- **Input**: set $A$ of applications which use the databases in set $D$ and a cost function for application testing
- **Output**: Partitions of databases into a set of migration waves $W$
- **Objective**: Minimize total application testing cost

Minimize: $f(W) = \sum_{k=1}^{[W]} \sum_{j: a_j \in A_{w_k}} t_j$

- **Constraint**: Size of each wave should not exceed $S_{max}$
DBMP is NP-hard

• SET-PARTITION can be reduced to DBMP

\[ S_{\text{max}} = \frac{1}{2} \cdot \sum_{s \in I} s \]

• Hyper-graph partitioning can be reduced to DBMP
  - Databases as nodes of hyper-graph
  - Applications as hyper-edges
  - Testing cost obtained using hyper-edge cuts
Proposed Solutions

• ILP based optimal solution for small problem instances

• Hyper-graph partitioning using hMETIS [U-MN]
  - May violate the wave size Smax constraint

• WAVE-FIT
  - Cost as good as hMETIS
  - Also honors the wave size constraint
Int. Linear Programming solution

• **Objective**

\[
\text{Minimize: } \sum_{k=1}^{W} \sum_{j=1}^{A} a_{jk} t_j. \tag{1}
\]

• **Constraints**

\[
\forall k, \sum_{i=1}^{D} s_i d_{ik} \leq S_{\text{max}}, \tag{2}
\]

\[
\forall i, \sum_{k=1}^{W} d_{ik} \geq 1. \tag{3}
\]

\[
\forall j, k, \sum_{i:d_i \in D_j} d_{ik} \leq |D_j| \cdot a_{jk}. \tag{4}
\]
ILP applicability limited

• Though optimal, useful for small problem instances only
hMETIS: hyper-graph partitioning tool

- Construction of a hyper-graph $H$ for a given DBMP
  - Databases as nodes of hyper-graph
  - Applications as hyper-edges
- Solving DBMP for $|W|$ migration waves is same as $|W|$-way partition of hyper-graph $H$
  - Weights of hyper-edge = particular application testing cost
  - Total cost = hyper-edge cuts + number of Apps
hMETIS...

- hMETIS provides good solutions
  - For many problems, cost is close to lower bound
  - But it may violate the wave-size constraint $S_{max}$
    - No strict upper-bound on partition size
  - Violation can in significant number of waves (upto 40%)
WAVE-FIT algorithm

• Sort (ascending order) applications based on number of databases used

• For each app $a_j$:
  - $g =$ set of DBs used by $a_j$
  - If $\text{sizeof}(g) < S_{\text{max}}$:
    • Repeat
      - Find an app $a_p$ such that $D_p$ has maximum overlap with $g$
      - If the combined size of $g$ and $D_p < S_{\text{max}}$ :
        » merge $g$ and $D_p$ to form the new $g$
    • Until $\text{sizeof}(g) < S_{\text{max}}$
  - Else:
    • Partition $g$ to satisfy the $S_{\text{max}}$ constraint
Evaluation

• Real-life problem
  - 191 apps, 116 DBs, 204 dependency edges
  - Both hMETIS and WAVE-FIT give optimal cost solution

• Experimental evaluation
  - Generated synthetic datasets based on real-life datasets
  - DB sizes
    • Lognormal (6.18, 2.79)
    • Uniform (10, 500000)
Comparison with naïve (bin-pack) approach
Comparison with naïve (bin-pack) approach
Testing cost using hMETIS and WAVE-FIT

\[ U_h = \text{testing cost due to hMETIS (Uniform case)} \]
\[ U_w = \text{testing cost due to WAVE-FIT (Uniform case)} \]
\[ L_h = \text{testing cost due to hMETIS (Lognormal case)} \]
\[ L_w = \text{testing cost due to WAVE-FIT (Lognormal case)} \]

Table 2: Ratio of Testing Costs by hMETIS and WAVE-FIT

<table>
<thead>
<tr>
<th>Experiment</th>
<th>( \frac{U_h}{U_w} )</th>
<th>( \frac{L_h}{L_w} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Experiment 5</td>
<td>1.01</td>
<td>1.00</td>
</tr>
<tr>
<td>Experiment 6</td>
<td>1.01</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Violation of wave size constraint by hMETIS
Violation of wave size constraint by hMETIS

(b) Violation of Wave Size Constraint by hMETIS
(Log-normal distribution)

- WaveSize Excess
- % TWSV of $S_{max}$

% of Violation

Number of databases

50 70 90 100 200 300 400
Summary

• Identification/Formalization of DBMP
  - a recurrent, important problem in the industry

• DBMP is NP-hard

• Three solutions
  - ILP based
  - Hyper-graph based
  - WAVE-FIT

• Characterization of realms where each of these solutions can be appropriate
Thanks!

Questions?