Design, Implementation, and Evaluation of a Tight Integration of Database and Workflow Engines

Peter Reimann, Holger Schwarz, Bernhard Mitschang

November 10th, 2011

Agenda

- Introduction

- Approach and prototype for a tight integration of database and workflow engines

- Evaluation of the tight integration

- Conclusion and future work
High-Level Architecture for Scientific Workflow Processing

Focus of this Talk

Provenance Framework

Provenance Capture/Management

Provenance Store

Workflow Execution Environment

Workflow Runtime

Local Data Processing in the Workflow System

Local Data Store

External Data Processing

External Resources/Applications

Proprietary Files

Database

Legacy Application
Local Data Processing in Workflows

- Kind of local data processing in workflows depends on kind of corresponding workflow language

- Data flow:
  - Local data processing based on pipeline and data stream processing
  - Several optimization opportunities already exist, e.g.:
    - Pipeline parallelism
    - High Performance Computing (HPC) solutions

- Control flow:
  - Local data processing based on handling workflow variables
  - Optimization opportunities have largely been neglected, in particular for data-intensive scientific workflows
Agenda

- Introduction

- Approach and prototype for tight integration of database and workflow engines

- Evaluation of tight integration

- Conclusion and future work
Tight Integration of Database and Workflow Engines

Conceptual architecture includes data processing optimizer to dynamically switch between both variants
Assignment Pushdown

- Completely execute variable assignments within database

![Diagram of Workflow Execution Environment with steps: 1) Submission of Notification, 2) Submission of Result Data, 3) Evaluation of Expression, 4) Assignment of Result]
Expression Evaluation Pushdown

- Forwards expression results back to workflow runtime
- Used for control flow decisions

![Diagram showing workflow execution environment and related processes]
Web Service Pushdown

- Avoids unnecessary transmissions of service results from workflow runtime to database system.
Literal Pushdown (not implemented)

- Avoids unnecessary transmissions of literal values from workflow runtime to database system
- Performed during workflow deployment
Agenda

- Introduction

- Approach and prototype for tight integration of database and workflow engines

- Evaluation of tight integration

- Conclusion and future work
Experimental Setting

■ Comparison of originalODE and ODE-TI in two scenarios:
  - Test of individual workflow activities to evaluate techniques in isolation
  - Test of a protein modeling workflow to evaluate techniques in combination
  - Same underlying data sets in both scenarios:
    – XML sequence of protein sequences with different data sizes (100 KB, 500 KB, 4 MB, 9 MB, 50 MB)
Effectiveness of Single Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Duration for ODE-Ti in % of original ODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment Pushdown (no expr.)</td>
<td>331</td>
</tr>
<tr>
<td>Assignment Pushdown (simple expr.)</td>
<td>227</td>
</tr>
<tr>
<td>Assignment Pushdown (complex expr.)</td>
<td>458</td>
</tr>
<tr>
<td>Expression Evaluation Pushdown (simple expr.)</td>
<td>190</td>
</tr>
<tr>
<td>Expression Evaluation Pushdown (complex expr.)</td>
<td>418</td>
</tr>
<tr>
<td>Web Service Pushdown</td>
<td>112</td>
</tr>
</tbody>
</table>

- 100 KB
- 500 KB
- 4 MB
- 9 MB
- 50 MB
## Results for Protein Modeling Workflow – Sequential Execution

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Average Duration in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 (100 KB)</td>
<td>11</td>
</tr>
<tr>
<td>199 (500 KB)</td>
<td>262</td>
</tr>
<tr>
<td>697 (4 MB)</td>
<td>291</td>
</tr>
<tr>
<td>1394 (9 MB)</td>
<td>1000</td>
</tr>
</tbody>
</table>

- **Main memory overload after ca. 200 iterations**:
  - Original ODE: 291 seconds
  - ODE-TI: 1000 seconds

- **Main memory overload after ca. 100 iterations**:
  - Original ODE: 4 seconds
  - ODE-TI: 31 seconds
Results for Protein Modeling Workflow – Parallel Execution

- **40 iterations (100 KB)**: 45 seconds
- **199 iterations (500 KB)**: 145 seconds
- **697 iterations (4 MB)**: 736 seconds
- **1394 iterations (9 MB)**: 1620 seconds

- Main memory overload after 17 minutes
- Main memory overload after 15 minutes
- Main memory overload after 10 minutes

Graph: Total duration of all workflow instances in seconds

Legend:
- Green: original ODE
- Blue: ODE-TI
Agenda

- Introduction
- Approach and prototype for tight integration of database and workflow engines
- Evaluation of tight integration
- Conclusion and future work
Conclusion

- Approach to improve workflow-local data processing by exploiting local database system whenever appropriate
  - In particular targeted at data-intensive workflows described in control-flow-oriented languages
  - Potential to improve both efficiency and reliability of such workflows

- Possible future work:
  - Implement of Literal Pushdown in ODE-TI prototype
  - Extension for pushing down more complex XQuery expressions
  - Improve scalability with respect to larger XML document sizes
  - Exploitation of main-memory-based database systems
THANK YOU FOR YOUR ATTENTION