Workflow Management in e-Science Analytics

Key Note Presentation at IWPLS’09 in Edinburgh

Peter Brezany
Institute of Scientific Computing
University of Vienna, Austria
brezany@par.univie.ac.at
Today’s Data Flood

Data Mining → Selection

Processing → Integration

Business

Simulations

Earth Observations

Scientific Experiments

Medicine
Motivation for Data Exploration

• **Cost aspects**
  – Cost of data producing and its persistent storage is high.
  – Data generators - satellites, particle accelerators, genome sequencing, supercomputing centers, etc. - collectively cost billions.
  – There is a need for retrieving vital information and knowledge from this data to compensate that great expense.

• **Scientific aspects**
  – This data is considered as an endless fuel for creativity.
  – Scientists want to apply it in innovative ways and novel combinations for discovery of new knowledge, validating hypothesis, understanding research phenomena, etc.
Data Storage Costs

• A single rack of storage equipment today:
  • 200TB of storage
  • 1200kg
  • 12kW

Reducing power consumption *(presented by Peter Kunszt)*

Crazy Ideas – or not?

• Build your integrated, data center in a ‘box’ highly optimizing the cooling air flow
• Build data center next to a green power plant (like a dam or geothermal)
• Build your data center where cooling is ‘free’
  • E.g. Greenland, Iceland
  • Build your data center on a ship – e.g. a decommissioned ocean liner
Solution: High Productive e-Science Analytics

• e-Science Analytics is a dynamic research field that includes rigorous and scientific methods of data preprocessing, integration, analysis, and visualization.

• A high productive analytics system is one that delivers a high level of performance, guarantees a high level of accuracy of analytics models and other results extracted from analyzed data sets while scoring equally on other aspects, like usability, robustness, system management, and ease of programming.

• The role of the Grid/Cloud/...
Knowledge Discovery in Grid Data Sets

- Grid supports knowledge discovery in:
  - Processing of large scale and distributed data
  - Utilizing distributed computing resources

- It offers
  - HPC resources
  - Massive storage resources
  - Advanced security

- Grid as an infrastructure in e-Science Projects:
  myGrid, DiscoveryNet, caBIG, Lead, TerraGrid, SEEK, GridMiner, DataMining Grid, ADMIRE and many, many others.
Outline

• Introduction and motivation
• GridMiner framework
  – DM platform
  – Workflow specification language
  – GridMiner portal
    • Workflow composition
    • Workflow steering
• Towards assisted workflow composition
• Support for provenance
• ADMIRE project approach
• Other ongoing and future work
GridMiner Project

• **Aim** - to provide tools and services for knowledge discovery in distributed and heterogeneous data sources on the Grid

• **Medical test applications**
  - prediction of the outcome of TBI patients
  - TCM – non-invasive blood sugar measurement

• **Implementations**
  - OLAP
  - Parallel and distributed algorithms
  - Management of distributed data
  - Workflow management & GUI

http://www.gridminer.org
Management of TBI patients

• **Traumatic brain injuries (TBIs)** typically result from accidents in which head strikes an object.

• The treatment of TBI patients is very resource intensive.

• The trajectory of the TBI patients management:
  – Trauma event
  – First aid
  – Transportation to hospital
  – Acute hospital care
  – Home care

• All the above phases are associated with data collection into databases – now managed by individual hospitals.
Knowledge Discovery Process in GridMiner

Data and functional resources can be geographically distributed – focus on **virtualization**.
Adapting the Data Mining Model Standard

CRISP-DM, SPSS

Virtual Organization

Service Provider

Service Provider

Service Provider

User

Data provider

Deployment

Evaluation

Data Preparation

Modeling

Data understanding

Data understanding

9/14/2009

Peter Brezany
High Level Architecture View

DM process (workflow) design and editing

DM process specification

DM process enactment

DM – data mining
Platform Design

DM services

workflow enactment
Goal: Optimizing predictive accuracy

$q_i$ - data quality  $m_i$ - modeling method
Service Parallelism Levels

Inter-Service Parallelism

Intra-Service Parallelism
Grid-Enabled Learning

• Based on the Black Board Architecture
Parallel Decision Tree Service

1. findBestSplitPoint()
2. partitionWinningAttr()

User Interface → DataMining-Service Master → Model

Site A

DataMining-Service Worker 1

1 2 3 4 5 6
AttrList A
P0 P1 P2

Site B

DataMining-Service Worker 2

1 2 3 4 5 6
AttrList B
P0 P1 P2
Parallel and Distributed Grid Services for Building Neural Network (NN) Classification Models

Implementation based on:

- Titanium (parallel Java dialect)
- Vienna Grid Environment
Structure of our Solution

Head-master

Sub-master0

Slave0
Slave1
Slave2

Data Distribution Scheme 1

Training Data for Sub-master0

Sub-master1

Slave0
Slave1

Data Distribution Scheme 2

Training Data for Sub-master1
Parallel NN Service
Parallel and Distributed NN Service

User Interface (Service Client)

Model

Master NN Service (Grid Service)

SlaveNN Service (Grid Service)

NNBuild (Master)

Process 1 (Slove)

Process 2 (Slove)

Set 1

Set X

...
Visualization

• Transformation of PMML documents to SVG
Parallel and Distributed OLAP Service

- Master
  - Virtual Cube
    - Slave 1
      - Sub Cube
        - Slave 2
          - Sub Cube
        - Slave 3
          - Sub Cube
    - Index Service
      - Indexes

Data → Virtual Cube → Slave 1 → Slave 2, Slave 3

Query answer XML
Supported SQL subset (important for DM):
SELECT column, [column] FROM table
WHERE condition [AND|OR] condition
ORDER BY column [,column] [ASC|DESC]
Collaboration of GridMiner(GM) - Services

Simple Scenario:

- GMDIS: Integration
- GMPPS: Preprocessing
- GMDMS: Data Mining
- GMPRS: Presentation

Data Sources → Intermediate Result 1 → Intermediate Result 2 (e.g. “flat table”) → Intermediate Result 3 (e.g. PMML) → Final Result
Collaboration

• Complex Scenarios:
Workflow Composition and Execution Models

Static Workflows

Dynamic Workflows

(a)

(b)
Workflow Management in GridMiner I

- **Dynamic Service Control Engine (DSCE)**
  - processes the workflow according to DSCL
  - Start, cancel, stop, resume, and change workflow operations

- **Dynamic Service Control Language (DSCL)**
  - based on XML
  - easy to use
GM Workflow language (DSCL)

User's view →

```
variables

composition

sequence
createService activityID="act1" ...

parallel
invoke activityID="act2.1" ...
invoke activityID="act2.2" ...

sequence
...```

Conversion to DSCL
Graphical User Interface
– End-User Level
### OLAP Query

**Dimension level**

- **Gender**
  - ANY

- **Year**
  - ANY

- **Month**
  - ANY

- **Day**
  - ANY

#### Working dimension:

**Location**

- Location
  - Change

#### Parameters:

<table>
<thead>
<tr>
<th>Country</th>
<th>City</th>
<th>Hospital</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>Wien</td>
<td>AKH</td>
<td>160.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LBH</td>
<td>240.0</td>
</tr>
<tr>
<td>Graz</td>
<td>Graz-AKH</td>
<td>190.0</td>
<td></td>
</tr>
<tr>
<td>Linz</td>
<td>Linz-LKH</td>
<td>16.0</td>
<td></td>
</tr>
<tr>
<td>Salzburg</td>
<td>Salzburg-LKH</td>
<td>140.0</td>
<td></td>
</tr>
</tbody>
</table>

- **Roll Up**
- **Drill Down**

- **Run Query**
- **Close**

---

**Time:** 13:06:35

**Actions:**

- Loaded Project: DEMO
Administration Level
(Data integration, Service Configuration, etc.)
New Development: Assisted Workflow Composition

• **Main features:**
  – Goal oriented tasks composition
  – Enumeration of available tasks/algorithms/parameters/workflows
  – Advanced service composition framework

• **Components**
  – Process Designer
  – Enactment Engine
  – Semantic Repository
  – Monitoring Tools
Composition Framework

- The framework should support
  - Service discovery
  - QoS negotiation
  - Sophisticated composition algorithm
  - Ranking of composite services
  - Grounding to concrete realization
Ranking of Composite Services

- **Aim** – selection of a composite service satisfying multiple constrains provided by a user.

- **Approaches**
  - aggregation of simple QoS parameters (e.g. cost)
  - computing of graph dependent aggregated parameters (e.g. overall execution time)
Composition Approaches

- **Manual**

  ![Diagram for Manual Composition Approaches]

- **(Semi-) Automated Passive**

  ![Diagram for (Semi-) Automated Passive Composition Approaches]
Composition Approaches

- Automated Active
Workflow Enactment Engine

• Requirements
  – Interactivity
  – Reliability
  – Security
  – Handling
  – Support for provenance
  – Autonomous behavior
Workflow Enactment Engine

• **Basic Characteristics**
  – WS-BPEL-compliant engine
  – Invocation:
    • WS-I
    • WSRF services
  – Running modes
    • standalone
    • Globus Toolkit service
  – Execution control
  – Monitoring
    • fully observable state of workflow activities and invoked services
  – Security
    • message and transport level
  – Database back-end

*Accepted as Globus Incubator Project; [http://dev.globus.org](http://dev.globus.org)*
Provenance Definition

• In art:
  – To find the owners of a work of art, tracing it from its present location and owner back to the hands of the artist.

• In computer science:
  – Provenance is metadata on the derivation history of data products. It provides information on the application used to derive a data product, and the inputs to that application.
Workflow Provenance

• **Our definition:**
  – “The process of collection of information on workflow execution in order to increase the trust of the user”.
    • Trust is achieved through transparency, execution, testing of others workflows.

• **Provenance categories:**
  – **Dataflow provenance:** Information on input(s) to the activities and output produced by workflow as well as by the individual activities.
  – **Control flow provenance:** Order of the execution of activities, name(s), time, and the number of activities.
Provenance architecture

*WD: Workflow Document
*PRMService: Provenance Recording & Monitoring Service
Provenance Visualization

1. Control flow visualization
2. Dataflow visualization
3. Explorer
4. XML Viewer
5. Data Viewer

Provenance Visualization

XML Viewer

Data Viewer

Explorer

Control flow visualization

Dataflow visualization

Provenance Visualization
ADMIRE Project

- ADMIRE = Advanced Data Mining Research for Europe
- CRISP-DM → CRISP-DKI

Architecture
- DMI Process Designer (component of the ADMIRE workbench)
- DMI Language (canonical workflow representation)
- DMI Gateway (DMI compiling, interpreting, workflow enactment)
- Put on top of OGSA-DAI

Pilot Applications
- Flood Management
- Customer Relationship Management
CRISP-DMI
DMI Process Designer (Developers: Ivan Janciak, Peter Brezany, et al. University of Vienna)
DMIL Code Example

```java
use eu.admire.Seq2Set;
use eu.admire.SeqTransform;
use eu.admire.SeqMerge;
use eu.admire.Quantize;
use eu.admire.MergeColumn;

function MergeAndCleanPattern-Seq (  
  PE (<Connection inData> => <Connection outData>) Filter,
  PE (<Connection inData> => <Connection outData>) Normaliser,
  PE (<Connection inData, Connection replacer> => <Connection outData>) MissingValuesFiller
) PE (<Connection inData1, inData2> => <connection outData>) {  
  Connection inputData1;
  Connection inputData2;

  /*! Initialize phase */
  SeqTransform transform = new SeqTransform();
  SeqMerge merge = new SeqMerge();
  Quantize quantize = new Quantize();
  Filter filter = new Filter();
  Normaliser normaliser = new Normaliser();
  MissingValuesFiller filler = new MissingValuesFiller();
  Seq2Set seq = new Seq2Set();
```
DMIL Code Example cont.

/* Transform phase */

inputData2 => transform.inData;

/* Merge phase */

inputData1 => merge.inData1;
transform.outData => merge.inData2;

/* Cleaning phase */

merge.outData => quantize.inData;
quantize.outData => normaliser.inData;
normaliser.outData => filter.inData;
filter.outData => filler.inData;
filler.outData => seq.inData;

return PE(<Connection inData1 = inputData1; Connection inData2 = inputData2> =>
<Connection outData = seq.outData>);
Visualization

Developers:
Marek Lenart, Ivan Janciak, P. Brezany
University of Vienna
Other Ongoing and Future Work

• Data management based on Dataspase paradigm
• Declarative DMI process specification
  – DMI query language
  – Query execution engine on top of OGSA-DAI
• (Semi-) Automatic support for workflow parameter optimization
  – Coupled to provenance activities