Scientific and Grid Workflow Management

Cesare Pautasso
University of Lugano
http://www.pautasso.info
Abstract

Grid workflow management systems coordinate multiple job submissions over heterogeneous Grid resources. They feature visual programming environments to give scientist a high-level view over distributed computations composed of Grid services.

This brief introduction to the field of scientific and Grid workflows includes a survey of selected workflow management tools and outlines current research trends.

Swiss Grid School SGS10, Lugano, Switzerland
Cesare Pautasso

Ph.D. at ETH Zürich (2004)
Post-Doc at ETH Zürich in the Systems (IKS) Group
  • Software: JOpera: Process Support for more than Web services
    http://www.jopera.org/
Researcher at IBM Zurich Research Lab (2007)
Assistant Professor at the new Faculty of Informatics, University of Lugano (USI), Switzerland (since September 2007)
  • USI Representative in the SwiNG Assembly (since 2007)
  • Grid Workflow Working Group Lead (since 2007)

More Information: http://www.pautasso.info/
Follow me on: http://twitter.com/pautasso/
Acknowledgements

Some material contained in this tutorial was adapted from slides originally published by:

Gustavo Alonso
Win Bausch
Ewa Deelman
Ian Foster
Yolanda Gil
Carole Goble
Roy Grønmo
Thomas Heinis
Rajesh Kalyanam
Francesco Lelli
Omer F. Rana
Heiko Schuldt
Frank Terpstra
Why Workflow Management on the Grid?
Kinds of Grid Computation

One Job Submission

Parameter Sweep
**Condition A**

- wet lab processing

**Condition B**

1 spot = 1 gene
- Expression level:
  - Green: A > B
  - Red: A < B
  - Black: A = B

- Cell population

---

Scientific and Grid Workflow (Cesare Pautasso)
in vitro  in silico

Condition A

wet lab processing

MicroArray Scanner

Extract raw spot intensities

Image processing

Determine Expression Pattern

Clustering

Data Preprocessing

Determine parameters for error model

Variability and error assessment

Significance assessment

likelihood for differential expression for each gene

Annotate and normalize data

1 spot = 1 gene
Expression level:
Green: A > B
Red:  A < B
Black: A = B

Condition B

Cell population

Time

Scientific and Grid Workflow (Cesare Pautasso)
Copy & Paste between different Websites
Copy & Paste between different Websites

Programming
Java, C++, C#, Fortran...

(Shell) Scripts
Tcsh, Bash, Makefiles, Python, Perl...

Workflows
Graphical, Drag & Drop and Connect Environments
Vision for Scientific and Grid Workflows

Make it easy to build Grid applications composed of multiple jobs

“Provide the scientist with a platform that takes care of all data handling and record keeping chores so that the user can concentrate on the science and not computer science.”
Workflow

GRID
## Some (Scientific) Workflow Management Systems

<table>
<thead>
<tr>
<th>Askalon</th>
<th>GWFE</th>
<th>Pegasus</th>
<th>Triana</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigbross Bossa</td>
<td>GWES</td>
<td>Pipeline Pilot</td>
<td>Trident</td>
</tr>
<tr>
<td>BioPipe</td>
<td>ICENI</td>
<td>P-GRADE</td>
<td>Twister</td>
</tr>
<tr>
<td>BPMN</td>
<td>Inforsense</td>
<td>PowerFolder</td>
<td>Ultimus</td>
</tr>
<tr>
<td>Breeze</td>
<td>JIGSA</td>
<td>Ptolemy II</td>
<td>Versata</td>
</tr>
<tr>
<td>Carnot</td>
<td>JOpera</td>
<td>Savvion</td>
<td>Viztrails</td>
</tr>
<tr>
<td>Con:cern</td>
<td>Kepler</td>
<td>Seebeyond</td>
<td>wftk</td>
</tr>
<tr>
<td>DAGMan</td>
<td>Karajan</td>
<td>SCIRun</td>
<td>XFlow</td>
</tr>
<tr>
<td>DiscoveryNet</td>
<td>Oakgrove's reactor</td>
<td>ScyFLOW</td>
<td>YAWL</td>
</tr>
<tr>
<td>Dralasoft</td>
<td>OSIRIS</td>
<td>SDSC Matrix</td>
<td>Wildfire</td>
</tr>
<tr>
<td>GEL</td>
<td></td>
<td>SHOP2</td>
<td>WFE</td>
</tr>
<tr>
<td>GridAnt</td>
<td>OSWorkflow</td>
<td>Taverna</td>
<td>WS-BPEL</td>
</tr>
<tr>
<td>Grid Job Handler</td>
<td>OpenWFE</td>
<td>Teuta (UML)</td>
<td>ZBuilder</td>
</tr>
</tbody>
</table>
Outline

Why Workflow Management on the Grid?

Discussion: Scientific vs. Grid vs. Business Workflows
  • Some Application Examples

Workflow Modeling Languages and Tools Overview
  • Grid Workflow Language Patterns

Running Workflows on the Grid
  • JOpera: Scientific Workflow for Eclipse
  • Workflows and Provenance
Scientific vs. Grid vs. Business Workflows
The Origins: Business Process Management

**who** has to do **what**, **when**
The Origins: Business Process Management

- A business process describes key procedures within an organization. They involve:
  - multiple steps
  - numerous people
  - large amounts of resources

- In large business organizations there are many factors that increase the complexity of the business processes:
  - processes are not well documented
  - conformance to rules not guaranteed
  - people lack information about context
  - company lacks monitoring tools
  - steps, people and resources are not properly coordinated

- Workflow Management Systems try to address these problems by automating the coordination aspects of a business process: who has to do what, when, and with which software tools.
Business Workflows

“

The automation of a business process where documents, information to be processed or tasks to be carried out are passed from one participant to another following a set of procedural rules.

”

Workflow Management Coalition (WfMC, 1993)
Scientific Workflows

“are networks of analytical steps that may involve, e.g., database access and querying, data analysis and mining, and many other steps including computationally intensive jobs submitted to high performance clusters and Grids”

Bertram Ludäscher
Modeling Workflows

User

Software Tool

Activity

Data

Control Flow

Data Flow

activity 1

activity 4

0

1

2

3

4

5

6

User

Software Tool

Activity

Data
Business Workflows

User

Activity

Control Flow

Software Tool

User

Activity

activity 1

activity 4

0

1

2

3

4

5

6
Scientific Workflows

Activity

Software Tool

Control Flow

Data

Activity

Software Tool

Data

activity

1

activity

4

0

2

3

4

5

6
Similarities:
Are scientists doing e-Business?

Capturing knowledge/best practices
Capture business processes within a company

Capture scientific experiments

Executable Models for Repeated Execution
Run a well defined procedure many times

Ensure that an experiment can be reproduced

Incorporate human decision in the process
Can we always do straight-through processing?

Hard to achieve full automation
Differences:
Do scientists need business transactions?

Rate of change
Changing business procedures requires management approval

*Exploratory scientific processes require high flexibility*

Which kind of data?
Travel reservations, Loan applications

*Large protein sequence databases, Astronomy image catalogs*

What is the ultimate goal?
Making profit

*Making science*
Scientific vs. Grid Workflows

Scientific workflows emphasize the design of virtual experiments:

- Data flow models
- Reusable “scientific computing” component library
- Interactive debugging, monitoring and steering
- Data provenance and lineage tracking for reproducibility
- Model versioning for exploratory customization

Grid workflows focus on the large-scale execution of scientific workflows:

- Mapping and adaptation to a dynamic run-time environment
- Provide access to shared workflows as a Grid service
- Parameterized Execution
- Centralized vs. Distributed Execution Architectures
- Fault Tolerance
- Optimization
Scientific Workflows on the Grid

• How can Scientific Workflow benefit from the Grid?
  
  1. Leverage underlying Grid middleware:
     • Resource Management
     • Job Scheduling for parallel Activities
     • Large Data Transfers (GridFTP) between Activities
  
  2. Improved QoS based on the workflow model
     • Grid resource reservation
     • Data replication
     • Data placement
     • Fault Tolerance
Example

http://www.jopera.org/
A Web Service-Enabled Workflow System for Climate Modeling Data Processing in TeraGrid

Rajesh Kalyanam
Lan Zhao
Taezoon Park
Sebastien Goasguen
Architecture

Application

Desktop Application

Web Portal

Workflow Engine

JOpera

Workflow Components

Data Components

Metadata Query

Data Discovery

Data Transfer

Data Transformation

Computation Components

Globus Job

Condor Job

Models / Tools

Purdue Data Management System

OPeNDAP

THREDDS

SRB/MCAT

Remote Data Proxies

Data Proxy

Remote Datasets

Remote Datasets

Computation Middleware

Condor-G

Globus

Computation Resources

Local Clusters

TeraGrid

Local Datasets

LARS

PTO

NWS

Climate Modeling

Remote Datasets

NWS

Climate Modeling

Computation Resources

Local Clusters

TeraGrid
This is a collection of multispectral and hyperspectral images, that are used for educational and research purposes at Purdue University. They are processed at the Laboratory for Applications of Remote Sensing.

Home Directory: `/demozone/home/lars.itap/LARSDATA/CDR_189`

File Details: `/demozone/home/lars.itap/LARSDATA/CDR_189/L7010904_022_032.LAN`
Workflow Model
Workflow Execution

From Rajesh Kalyanam
Workflow Results

From Rajesh Kalyanam
Workflow Lifecycle
Workflow Lifecycle

Scientific Computation → Modeling → Workflow Model → Log Analysis → Workflow Instances → Simulation → Scientific Computation

From Gustavo Alonso
Workflow Modeling Methodologies
Bottom up Composition

1. Select components from a library
   a. Lookup services in a public registry
   b. Import from external Web service (WSDL)
   c. Search the standard library

2. Build a workflow using a drag, drop and connect **modeling** environment

3. Run, Test, and Debug the execution **within the same modeling environment**

4. Share and Publish it as Web Service
Top down Decomposition

1. Define a **goal** and Draw a *skeleton of the workflow* that satisfies it

2. Refine it and **Bind** services into it:
   - Search for existing matching services
   - Build missing services (if necessary)
   - Add required data transformations

3. Run, Test, and Debug the execution **within the same modeling environment**

4. Share and Publish it as Web Service
Iterative Composition

Change, Rediscover
Build New services

Discover services

Refactor

Model Service Composition

Manage

Deploy Run, Test

Compile

Check
Workflow Modeling Languages and Tools Overview
HeNCE - The Ancestor of Grid Workflows?

Taverna Workbench v1.7.1.0

Taverna

- id
- namespace
- Object
- getDragonSimpleAnnotatedImages
- getJpegFromAnnotatedImage
- Parse_Moby_Data_JPEGImage
- Decode_base64_to_byte
- Parse_Moby_Data_SimpleAnnotatedJPEGImage

Workflow Outputs:
- images
- annotations
VizTrails
OSIRIS
JOpera

Scientific and Grid Workflow (Cesare Pautasso)
Grid Workflow Language Patterns
### Workflow Pattern Variants

1. **Simple Parallelism**
   - Implicit
   - Explicit

2. **Data Parallelism**
   - Static
   - Dynamic

3. **Pipelining**
   - Best Effort
   - Blocking
   - Buffered
   - Superscalar
   - Streaming
   - Hybrid
   - Synchronized
   - Out of Order
Modeling Simple Parallelism

Data Flow, Graph Based

SCIRun
Kepler
Triana
Modeling Simple Parallelism

Control Flow, Graph Based

![Diagram showing simple parallelism with control flow and graph-based modeling using JOpera and GEL on the left, and UML on the right.](image)
Modeling Simple Parallelism

Control Flow, Block Based

BPMN

Sequence

Flow

WS-BPEL
Modeling Data Parallelism

Data Flow, Graph Rewriting

Static or Dynamic

Triana  Taverna  JOpera
Modeling Data Parallelism

Control Flow, Block Based, Dynamic

Sequence

ParallelForEach

T

WS-BPEL
AGWL
Karajan
GEL
Modeling Pipelined Execution
Pipelining Semantics
Best Effort Pipelined Execution

Drop data elements on pipeline collisions

Advantages:
- Simplified implementation
- Some applications may tolerate data loss

Problem:
- Downsampling is non deterministic
Blocking Pipelined Execution

Tasks are blocked if successors are busy

Advantages:
  • Avoid data loss in the pipeline

Problem:
  • Pipeline speed limited by slowest task
  • Data may be lost before it enters the pipeline
Buffered Pipelined Execution

Tasks are decoupled by buffers

Advantages:
- Collisions are prevented
- Best applied to tasks having variable speed

Problem:
- Buffer capacity is limited
  (Blocking still needed – Hybrid semantics)
Streaming Pipelined Execution

Tasks exchange data while running

Advantages:
- Suitable for a distributed (P2P) engine

Problems:
- Shifts complexity from the workflow engine to the tasks
- Tasks exchange data while running
- Workflow/Task interface more complex
Running Workflows on the Grid
Workflow Model

Act 1
Act 2
Act 3
Act 4
Act 5
Act 6
Act 7

Workflow Management System

Workflow Users
Workflow Participants

Workflow Modelers

Adapters

Scientific Software Developers

Grid Schedulers

Grid Resources

Wrapper Developers
Standard APIs

Process Definition Tools

Workflow API and Interchange formats

Workflow Enactment Service

Workflow Engine(s)

Invoked Applications

Workflow Client Applications

Administration & Monitoring Tools

Interface 1

Interface 2

Interface 3

Interface 4

Interface 5

From WFMC, Workflow Reference Model, 1998
Wrappers and Grid Applications

Workflow Management System

Worklist Handler

Grid Scheduler

Application X

adapter

wrapper

Application X

Application X

Application X

Act X
Wrappers and Legacy Applications

• The workflow engine is also in charge of connecting the different scientific applications.
• These applications do not have to talk directly to each other, they do it through the workflow engine.
• Most engines target a service oriented applications for which they provide very good connectivity through standardized protocols. Otherwise, the interface adapters must be developed on a case by case basis (as a last resort manual integration may be required!)
• For legacy application, a wrapper must be built so that the workflow engine can communicate with the application. The wrapper can be a simple relay of commands and data, or a complete translation program implementing functionality not present in the legacy application.
• For most Grid applications, the interaction takes place through a Grid scheduler, which is responsible for managing the distributed execution of the applications.
Run-time Abstraction Levels
Run-time Abstraction Levels

- A design-time workflow model needs to be mapped across different abstraction levels in order to be executed at run time.
- User request the execution of a new workflow instance.
- The abstract workflow is mapped to an executable instance by:
  - Finding suitable service implementations and binding them to the tasks
  - Rewriting the workflow graph based on a set of refinement rules
  - Planning required data staging, registration, placement, replication and transfer operations
- Each task of the resulting executable workflow is then submitted to a Grid resource manager so that it can be scheduled on suitable resources
- The mapping can be done:
  - when the workflow is started at instantiation time (statically)
  - incrementally as the workflow runs (adaptive execution with dynamic late binding)
Example: Binding with WS-BPEL
Workflow Binding Lifecycle

- Library Registration time (*classification*)
- Modeling time (*static early binding*)
- Compilation time (*blacklisting*)
- Deployment time (*customization*)
- Startup time (*testing*)
- Task Execution time (*dynamic late binding*)
- Failed invocation time (*rebind on retry*)
JOpera
Scientific Workflow for Eclipse

http://www.jopera.org/
- **High Level Workflow Language**
  - Data and Control Aspects (Visual Representation)
  - Recursion, Iteration, Parallelism and Pipelining

- **Open and Extensible Component Model**
  - Run existing code without changes
  - Synchronous, Asynchronous, and Streaming interaction
  - Web services support (Axis, WSIF, REST)
  - Secure access to remote file systems and hosts (SSH)
  - Easy to integrate with existing schedulers (e.g. Condor)
- High Level Workflow Language
  - Data and Control Aspects (Visual Representation)
  - Recursion, Iteration, Parallelism and Pipelining

- Open and Extensible Component Model
  - Run existing code without changes
  - Synchronous, Asynchronous, and Streaming interaction
  - Web services support (Axis, WSIF)
  - Secure access to remote file systems and hosts (SSH)
  - Easy to integrate with existing schedulers (e.g. Condor)

- Strong Eclipse Foundation
  - Platform Independent (Eclipse/Java)
  - Flexible, Extensible, Modular and Embeddable
JOpera Visual Composition Language

Workflows are modeled using multiple viewpoints:

1. Data Flow Graph

2. Control Flow Graph

3. Service Bindings
JOpera Example: Doodle Map Mashup

Setup a Doodle with Yahoo! Local search and visualize the results of the poll on Google Maps
Doodle Map Mashup Architecture

Web Browser → Workflow Engine → RESTful Web Services APIs

RESTful API

GET

POST

GET

Workflow Engine

GetYahooLocal

ConvertY2D

PostDoodleCreatePollText

ParsePollId

GetPoll

Wait

ShowGMapDoodle

24.6.2010 Scientific and Grid Workflow (Cesare Pautasso)
DoodleMap with JOpera

Poll: hamburger

CP has created this poll.

10001

<table>
<thead>
<tr>
<th>Gray's Papaya</th>
<th>Gray's Papaya</th>
<th>Hilton Millenium</th>
<th>Island Burgers &amp; Shakes</th>
<th>Le Parker Meridien New York</th>
<th>Corner Bistro</th>
<th>ESPN Zone</th>
<th>Warwick New York Hotel</th>
<th>The Jekyll &amp; Hyde Club</th>
<th>Dow the Hatc</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Extensible JOpera Component Model

Combine in the same workflow jobs implemented using an open and extensible set of technologies

JOpera Workflow

- WSDL
- Java
- Human
- XML
- SQL
- SSH
- Condor

- Snippets
- Methods
- XSLT
- XPath
Sharing Workflows as a Service

JOpera processes are automatically published to clients using a variety of access protocols.

Web Clients

WS Clients

Eclipse RCP Clients

REST

WSDL

Java

JOpera Workflow

WSDL

Java

Human

XML

SQL

SSH

Condor
JOpera ARC Integration Demo

Scientific and Grid Workflow (Cesare Pautasso)
Workflows and Provenance
Lineage in Scientific Workflows

Scientists consider the “capture and generation of provenance information as a critical part of the workflow-generated data”

“Sharing workflows is an essential element of education, and acceleration of knowledge dissemination.”

Ewa Deelman et al.
Where does this picture come from?

This photo was taken July 21, 1981, when the Voyager 2 spacecraft was 33.9 million km from the Saturn planet.
Title: White Arabian Horse

METADATA

Date: 16.4.2005
Dimension: 640x480
Colors: 32bits
Size: 1.2MB
Format: JPEG

FOR SALE
Would you buy a horse without this?

American Morgan Horse Register

This Certifies that the Morgan MARE

Named BIT-O-WY RUBY

No. 00146051 Foaled May 18, 1998

"Qualified By Blood Testing"

Color BAY Marked CONNECTED STAR, STRIP, SNIP. BLACK EYES.

Bred by DON BOSMAN, CHEYENNE WY
Lineage in Spreadsheets

<table>
<thead>
<tr>
<th>A</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>01-Jan</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>01-Feb</td>
<td>20</td>
<td>-50</td>
</tr>
<tr>
<td>E</td>
<td>01-Mar</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>F</td>
<td>01-Apr</td>
<td>40</td>
<td>77</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **F19** = SUM(E14:G15)

<table>
<thead>
<tr>
<th>01-Jan</th>
<th>01-Feb</th>
<th>01-Mar</th>
<th>01-Apr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-50</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>-60</td>
<td>77</td>
</tr>
<tr>
<td>178</td>
<td>40</td>
<td>11</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.90994</td>
<td>52.6205</td>
<td>43.55361</td>
</tr>
<tr>
<td>36.36135</td>
<td>58.67424</td>
<td>60.91148</td>
</tr>
<tr>
<td>43.55361</td>
<td>53.04644</td>
<td></td>
</tr>
<tr>
<td>113.0464</td>
<td>6.953557</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.273</td>
<td>26.273</td>
<td>38.273</td>
</tr>
<tr>
<td>-18.0929</td>
<td>-7.09289</td>
<td>4.907114</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>59.55324</td>
</tr>
</tbody>
</table>
# Lineage in Spreadsheets

![Spreadsheet Image]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>01-Jan</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>01-Jan</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>01-Feb</td>
<td>20</td>
<td>-50</td>
<td>20</td>
<td>01-Feb</td>
<td>-88</td>
</tr>
<tr>
<td></td>
<td>01-Mar</td>
<td>30</td>
<td>30</td>
<td>-60</td>
<td>01-Mar</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>01-Apr</td>
<td>40</td>
<td>77</td>
<td>40</td>
<td>01-Apr</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>01-Jan</td>
<td>100</td>
<td>67</td>
<td>11</td>
<td>178</td>
<td>01-Jan</td>
</tr>
<tr>
<td></td>
<td>2005</td>
<td>12.90994</td>
<td>52.6205</td>
<td>43.55361</td>
<td>36.36135</td>
<td>2005</td>
</tr>
<tr>
<td></td>
<td>2006</td>
<td>214.3613</td>
<td>141.6387</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2005</td>
<td>72.7227</td>
<td>88</td>
<td>99</td>
<td>111</td>
<td>106.0929</td>
</tr>
<tr>
<td></td>
<td>2006</td>
<td>15.2773</td>
<td>26.2773</td>
<td>38.2773</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2007</td>
<td>-18.0929</td>
<td>-7.09289</td>
<td>4.907114</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Lineage in Spreadsheets

#### Example Data

<table>
<thead>
<tr>
<th>Date</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Jan</td>
<td>10</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>01-Feb</td>
<td>-50</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>01-Mar</td>
<td>30</td>
<td>30</td>
<td>-60</td>
</tr>
<tr>
<td>01-Apr</td>
<td>40</td>
<td>40</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Jan</td>
<td>-8</td>
<td>57</td>
<td>11</td>
</tr>
<tr>
<td>01-Feb</td>
<td>52.6205</td>
<td>36.36135</td>
<td>43.55361</td>
</tr>
<tr>
<td>01-Mar</td>
<td>113.0464</td>
<td>6.953557</td>
<td></td>
</tr>
</tbody>
</table>

#### Lineage Diagram

The lineage diagram shows the dependencies and calculations in the spreadsheet. Each cell is linked to its parent cells, illustrating the flow of data and calculations. The values are calculated using formulas, as indicated in the cells.
## Lineage in Spreadsheets

![Spreadsheet Image]

### Table Example

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>-50</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>30</td>
<td>30</td>
<td>-60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>77</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>67</td>
<td>11</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12.90994</td>
<td>52.0203</td>
<td>43.55001</td>
<td>-66.36135</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>214.3613</td>
<td>141.6387</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>72.7227</td>
<td>88</td>
<td>99</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15.2773</td>
<td>26.2773</td>
<td>38.2773</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-16.0929</td>
<td>-7.09289</td>
<td>4.07114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lineage in Databases

What is the relationship between these tuples?

Problem: Query Inversion
Lineage in Software Development

What’s in a Makefile?

CC = gcc
CFLAGS = -Wall -g

program: main.o input.o output.o logic.o
  $(CC) $(CFLAGS) main.o input.o output.o logic.o -o program

main.o: main.c input.h output.h logic.h
  $(CC) $(CFLAGS) -c main.c
input.o: input.c input.h
  $(CC) $(CFLAGS) -c input.c
output.o: output.c output.h
  $(CC) $(CFLAGS) -c output.c
logic.o: logic.c logic.h
  $(CC) $(CFLAGS) -c logic.c
Lineage in Software Development

Where does my program come from?

CC = gcc
CFLAGS = -Wall -g

program:  main.o input.o output.o logic.o
  $(CC) $(CFLAGS) main.o input.o output.o logic.o -o program

main.o:  main.c input.h output.h logic.h
  $(CC) $(CFLAGS) -c main.c

input.o:  input.c input.h
  $(CC) $(CFLAGS) -c input.c

output.o:  output.c output.h
  $(CC) $(CFLAGS) -c output.c

logic.o:  logic.c logic.h
  $(CC) $(CFLAGS) -c logic.c
Lineage in Scientific Workflows

An ideal scientific workflow should document all of the steps linking the original observations with the final published results so that the process can be reproduced.
Data Provenance

Where does this output document come from?
Change Propagation

What to recompute if this input changes?
Conclusion
Reuse

Data Products

Adapt, Modify

Data, Metadata, Provenance Information

Execution

Distributed

Compute, Storage and Network Resources

Map to available resources

Mapping

Resource, Application Component Descriptions

Workflow and Component Libraries

Workflow Template

Populate with data

Workflow Instance

Modeling

Data, Metadata Catalogs

Executable Workflow

Map to available resources

From Ewa Deelman
e-Science as Workflow?

Provenance Query

What I Did

What I Am Doing

What I Want to Do

Execution environment

Schedule

Executable

Executing

Executed

Not yet executable

From Ian Foster
Some References

OGF Workflow Research Group
http://www.isi.edu/~deelman/wf–rg/
Download This Tutorial Material
http://www.pautasso.info/lectures/sgs10workflow.pdf
Free Download

http://www.jopera.org/
http://www.pautasso.info/lectures/sgs10workflow.pdf