UniProv

A flexible Provenance Management System for UNICORE

5. Oct. 2016 | André Giesler (a.giesler@fz-juelich.de), Myriam Czekala, Björn Hagemeier, Richard Grunzke
Jülich Supercomputing Centre (JSC)
Outline

• **Overview** on Data Provenance

• **The motivation:** Adding interoperable provenance tracking to UNICORE-based workflows

• **UniProv** Provenance management system for UNICORE, “look behind the scenes”

  ▪ **Acknowledgment** and special thanks to LSDMA Project - www.helmholtz-lsdma.de
Provenance

Definitions

- Latin: “provenire“
- The place of origin or earliest history of something
- The beginning of something’s existence
- A record of ownership of a work of art, used as a guide to authenticity or quality
- Related terms: lineage, genealogy, pedigree, …
Computational Provenance

- **Origin** and **processing history** of an artifact
  - Usually: data products, figures, …
  - Sometimes: workflow & script evolution …
- Provenance sightings and issues
  - Data-intensive science
  - Bioinformatics, climatology, …
  - Scientific workflows & scripts …
  - Serious problems of research reproducibility
    - 70-90% of experiments cannot be reproduced
    - 50% of scientists failed to reproduce their own experiments
- Standards and storage management
- Privacy vs. provenance
Use Cases for Provenance

- Audit trail: trace data generation and possible errors
- Attribution: determine ownership and responsibility for data and scientific results
- Data quality: from quality of input data, computations
- Discovery: enable searching of data, methodologies and experiments
- Replication: facilitate repeatable derivation of data

⇒ Reproducible Science
Scientific Workflows: ASAP!

Automation
workflows to **automate** computational aspects of science

Scaling
(exploit and optimize machine cycles)
make use of parallel compute resources
be able to handle large data

Abstraction
Evolution, Reuse (human cycles)
should be easy to (re-)use, evolve, share

Provenance
workflows should capture processing history, data lineage
=> traceable data- and wf evolution
Adding interoperable provenance tracking to UNICORE-based workflows
UNICORE

- Federation Middleware System
- Main characteristics
  - Abstract, secure, and intuitive access to supercomputing resources
  - Mature security mechanisms for distributed infrastructures (X.509, SAML, XACML, OpenID)
  - Different clients (command line, graphical, web based access and RESTful APIs)
  - Workflow support
- Areas of application
  - Human Brain Project
  - 3D-PLI workflow (neuroimaging)
Use Case – 3D PLI Workflow

- Scanning and studying nerve fibers of human brain sections at the micro-meter level
- Complex Image Processing to 3D map
- Track lineage of the final data products
  - Capture used parameters, libs, software versions, supercomputers
  - Which application versions used in workflow B? List all parameter values used by task t of the workflow.
- Adding and querying user annotations
Research objectives

- Enabling traceability to UNICORE-based jobs and workflows
- Building an interoperable provenance framework
  - For easy inter-change of provenance information
  - To adapt various provenance providers
- Storage management for provenance information
  - Security and consistency
  - Optimized data model for querying the provenance graph
UniProv
A Provenance Management System for UNICORE
Using a common model

W3C has published the ‘PROV’ family of recommendations

Provenance is a record that describes the people, institutions, entities, and activities, involved in producing, influencing, or delivering a piece of data or a thing in the world.
Using a common model

Example: Book writing activity

- Author
  - wasAttributedTo
  - wasAssociatedWith
- Book
  - wasGeneratedBy
- Book Writing
Using a common model

Example: Mapping of UNICORE-triggered image processing
Extending W3C-PROV

- very generic, universal
- Core data model (PROV-DM)
- Serializations available
  - PROV-O: OWL2 ontology
  - PROV-XML: defines XML schema for PROV-DM
  - PROV-N: human-readable notation
  - PROV-JSON(-LD): lightweight extensible representation
ProvONE Workflow extension
UniProv Implementation Details

- Capturing runtime information from UNICORE job management services and workflow engine by server-side integrated Java provenance-loggers
- Using ontologies (OWL2) for encoding semantically the provenance model
  - PROV Ontology (PROV-O) to encode the PROV Data Model
  - PROVONE – to encode the workflow trace and runtime logic
  - UniProv Ontology – to encode UNICORE domain specific axioms as user annotations, logical structures (loops), resource parameters (cores, memory per job)
- Apache Jena Ontology API to serialize the provenance data
UniProv Provenance Storage

- Different options:
  - RDBMS
  - Graph Database
- Natural mapping from directed provenance graph
- neo4j graph database chosen as standard repository for UniProv
  - Requires mapping from the PROV encoded output to property graph model
  - Declarative easy-to-use query language (Cypher)
  - Visualizing and traversing graphs by neo4j web interface
Mitglied der Helmholtz-Gemeinschaft

Unicore2PROV

Module

Ontologies

Predefined PROV Patterns

PROV

PROV One

Uni Prov

PROV-O Representation (Turtle)

Interoperability

PROV2Neo4j Generator

PROV Toolbox

Neo4j API

Provenance Graph

Backend

3rd Party Web Service

Triple Store

query

User Interface

Neo4j Browser

Command Line

Uniprov - Provenance Management System

capture provenance

generate

use

use

PROV-JSON

Turtle-RDF

Mitglied der Helmholtz-Gemeinschaft
PROV-encoded output of Unicore Job (PROV-N)

document

prefix provone <http://purl.dataone.org/provone/2015/01/15/ontology#>
prefix owl <http://www.w3.org/2002/07/owl#>
prefix rdfs <http://www.w3.org/2000/01/rdf-schema#>
prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix uniprov <http://unicore.eu/ns/uniprov#>
prefix pre_1 <file://stdin/usr/local/unicore/UNICORE_Servers-7.6.0/FILESPACE/29bae22d--f28025f86b72/>
prefix pre_0 <file://stdin/>

wasEndedBy(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, -, -, 2016-08-30T16:35:14.330000+00:00)
wasAssociatedWith(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, pre_0:unicore, -)
wasAssociatedWith(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, pre_0:unicore, pre_1:input)
used(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, pre_0:environmentVariables, -)
used(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, pre_0:environmentResources, -)

hadMember(pre_0:environmentResources, pre_0:Job_Type)
hadMember(pre_0:environmentVariables, pre_0:VARIABLE_A)
wasStartedBy(pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, -, -, 2016-08-30T16:34:12.586000+00:00)
entity(pre_0:userAnnotations, [prov:type='prov:Collection', prov:type='uniprov:JobAnnotationsCollection'])
entity(pre_0:environmentResources, [prov:type='uniprov:JobResourcesCollection', prov:type='prov:Collection'])
entity(pre_0:environmentVariables, [prov:type='prov:Collection', prov:type='uniprov:EnvironmentVariablesCollection'])
wasGeneratedBy(pre_1:stderr, pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, -)
wasGeneratedBy(pre_1:stdout, pre_0:29bae22d-4ba2-492c-a093-f28025f86b72, -)

endDocument
Visualized PROV-modeled output
Summary and Outlook

- UniProv provides provenance support for UNICORE job and workflow management
- Enabled interoperability by using a standard provenance data model
- Offers a natural storage and query mechanism for provenance data

- Ongoing implementation process
  - Finished PROV modeling of single compute jobs
  - Mapping the complex UNICORE workflow engine to ProvONE and UniProv extension
- Integration in HBP infrastructure
Thank you for your attention!