CS 595 - Hot topics in database systems:
Data Provenance

II. Beyond Database Provenance
II.1 Workflow Provenance

Boris Glavic

November 28, 2012

Workflow Provenance

- Track back the execution of workflows
- Usually developed for a specific workflow management system
- Mostly coarse-grained
  - Workflow "operations" as black boxes
- New challenges
  - Do not know provenance behaviour of atomic operations
  - Probably no fixed data model
  - Distributed storage and execution models with limited control

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

Excursion Workflows

What is a Workflow?

- A collection of simple(r) tasks such as
  - Filter out certain wavelengths from an image
  - Remove duplicates from a data set
  - Compute the mean of a dataset
  - Plot some data as a graph
  - ...
- ...Combined to achieve a more complex task
  - Simulating the changes of ocean currents for the next 100 years and visualize the result
  - Analyze raw telescope input data to detect and classify stars
  - ...

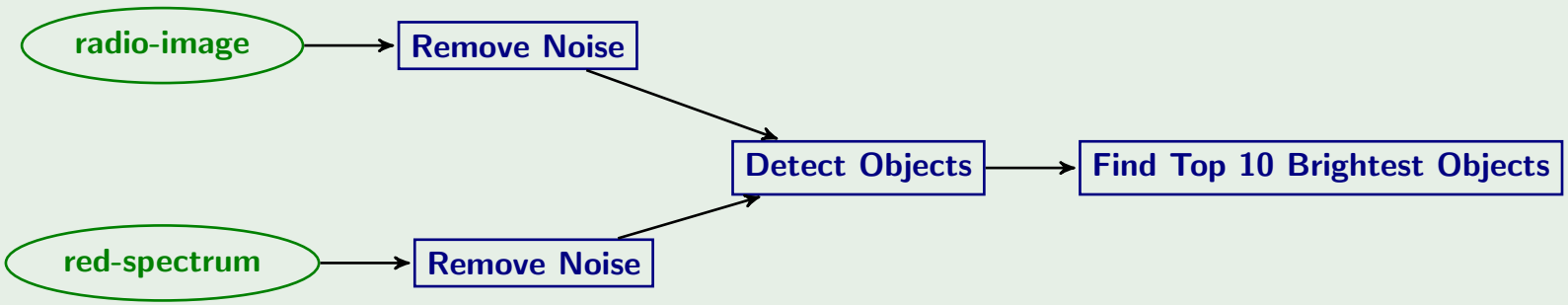
GY

Excursion Workflows

How to model a Workflow?

- Usually a directed (acyclic?) graph (DAG)
- Nodes are tasks
- Edges denote data-flow
  - . . . sometimes also control flow

Example



Excursion Workflow Management Systems

What is a Workflow Management System (WMS)?

- "Supervised" execution environment for workflows
- Usually comes with **workflow** and **execution** model
  - maybe also **data** model
- Visual interface for constructing workflows

ILLINOIS INSTITUTE OF TECHNOLOGY

Excursion Workflow Management Systems

Why are WMS needed?

- Reuse of components (tasks)
- Reuse of workflows
- Documentation of what happened and preservation of results
  - $\Rightarrow$provenance!
- Hide complexity
  - . . . of distributed processing
  - . . . of scheduling tasks
  - . . . of monitoring execution
  - . . . of data movement
  - . . . of fault tolerance

Excursion Workflow Management Systems

Some WMS

- Kepler
- Taverna
- Triana
- Pegasus
- GridNexus
- DiscoveryNet
- BioWBI
- GridBus
- ...

GY

Excursion Workflow Management Systems

Execution Models

- Grid-based: Tasks are Web-service calls
- MapReduce
- Traditional processes
- ...

Scheduling

- User-defined vs. automatic
- Strategies:
  - minimize execution time
  - minimize resource usage

Excursion Workflow Management Systems

Data Models

- No model: data stored in flat files
- XML-based: Kepler (one option)

Fault Tolerance

- Checkpoints
- Redundancy
- Replication

Data Movement

- User-defined
- Automatic

Excursion Workflow Management Systems

Example (Taverna)

- Workflows for Biologists
- Part of the *my*Grid project
- Tasks are services (mostly web-services)
- Taverna consists of
  - User-interface: design workflows, discover available tasks, browse provenance
  - Workflow enactor: engine to execute workflows using the web-services
  - Discovery of services (service registry and ontologies)
  - Resolving interface mismatches
    - Ontologies for semantics and structure of service input and outputs
    - "shams": components to convert data for structural mismatches

Outline

Overview

Approaches

- WMS are very diverse
- ⇒Almost one approach per system
- Open Provenance Model
  - A graph model for provenance
  - Community design
  - Should enable interoperability between systems
- Most provenance models are coarse-grained
  - ⇒tasks are black-boxes
  - Or provenance behaviour has to be supplied as part of task descriptions
- WMS as supervised execution environment that tracks provenance

Kepler - Efficient Provenance Storage and Querying

Kepler

- Data model and Execution model:
  - Tasks create and consume XML data
  - Input of a task is a stream (sequence) of XML elements (tokens)
    - Tokens are either data or collections
    - Each token has a unique identifier
  - A task can
    - Pass on input tokens unmodified
    - Add new tokens
    - Delete tokens
    - Modify tokens
- Tasks expected to output dependency information (provenance)
  - A token x depends on a token y through task a

Kepler cont.

Provenance storage

- Dependencies are stored in central DB
- Rules to derive additional annotations
- Compression schemes for provenance
  - Store overlapping provenance only once
  - Do not materialize provenance that can be derived through rules

ILLINOIS INSTITUTE OF TECHNOLOGY

Taverna

- Tasks are web-services
- Ontologies describe services and their interfaces
- Automatic discovery of services and semantic and structural mismatches
- Provenance is represented using RDF (resource description framework) and ontologies RDFS (RDF schema)
  - RDF: A graph spanned by triples (`object`, `predicate`, `subject`)
  - Nodes are entities
  - A triple is an labelled edge from `object` to `subject` labelled with `predicate`
- Querying using RDF query languages like SPARQL

Taverna

Different views on provenance

- Process view: Event log (Process provenance)
- Data view: Data provenance
- Organizational view: Track the who (which machine, user, ...)
- Knowledge view: Using the ontologies to provide abstraction
  - e.g., `instanceOf` for data products
- Connection between provenance graphs for workflows
  - Using URIs to identify data, processes, and invocations

ILLINOIS INSTITUTE OF TECHNOLOGY

PigLatin with Provenance for Workflows

PigLatin

- SQL-like data flow language
- Compiled into MapReduce tasks

Provenance for PigLatin Workflows

- Tasks = Modules
  - Written as PigLatin scripts
- Limit to subset of language
- Use provenance polynomials (the graph representation)
  - Plus aggregation extensions
- ⇒fine-grained provenance but tasks have to be expressed in Pig

Generalized MapReduce Workflows (GMRW)

- Presented by Jiawei
- Workflows tasks are map and reducer functions
- plus union and split to deal with multiple outputs or inputs
  - MapReduce only supports single output and input data sets
- Store provenance using the `(key,value)` model of MapReduce
  - Add provenance information to the `value` part
- Replace the original mapper and reducer functions of the workflow with Wrappers
  - A wrapper strips of the provenance information
  - Calls the original function
  - Attaches provenance information to the output of the function
  - $\Rightarrow$Provenance of these functions derived from generic mapper/reducer semantics

Open Provenance Model (OPM)

- Approach to standardize provenance models
  - . . . a bit tailored towards workflow mindset
- Abstract graph based model
- Provenance graph models past execution of a workflow execution and interdependencies

ILLINOIS INSTITUTE OF TECHNOLOGY

Open Provenance Model (OPM)

Node Types

- Artifact: Piece of data or physical state
  - E.g., a blood sample, an image file
- Process: Transformation producing and/or consuming artifacts
  - E.g., image transformation, a clustering algorithm, a weather sensor measurement process
- Agent: An entity controlling or triggering a process
  - E.g., a user starting the workflow, the workflow enactment engine,

Open Provenance Model (OPM)

Edges

- Denote causal dependencies
- Edge Types
  - Process → Artifact: Process created artifact
    - ⇒Outputs
  - Artifact → Process: Artifact was used by process
    - ⇒Inputs
  - Artifact → Artifact: Artifact was derived from artifact
    - data provenance as we covered in this class
  - Agent → Process: Agent controlled process
    - E.g., user started the process, workflow engine executed process
  - Process → Process: Process was trigged by process
    - E.g., process starts several subprocesses, .e.g., divide-and-conquer algorithm

Open Provenance Model (OPM)

Reasoning Rules

- Generic inference rules applicable to all OPM graphs
- Transitive closure of, e.g., artifact-artifact edges
  - E.g., $\exists(A_1, A_2) \wedge (A_2, A_3) \Rightarrow (A_1, A_3)$
- Derive new process-artifact and artifact-process edges based on sub-processes (process-process) edges
  - E.g., $\exists(P_1, P_2), (P_2, A) \Rightarrow (P_1, A)$

ILLINOIS INSTITUTE OF TECHNOLOGY

Karma 3

- Grid-model (workflow steps are services)
- Application Factory Toolkit: wraps programs as seb services
- Services used in workflows can be workflows (hierarchical composition)
- Variation of Business Process Execution Language (BPEL) to describe workflows
- A provenance service collects provenance data (messages)
- Both services and the workflow enactment engine are instrumented to generate provenance messages
  - Two types of models: notification model (asynchronous) and direct API service calls (synchronous)
- The provenance service stores incoming messages in a relational DB

Karma 3

Querying provenance

- provenance service API calls
- SQL over the backend DB
- Hardcoded client-side queries (e.g., Java)

ILLINOIS INSTITUTE OF TECHNOLOGY

PReServ: Provenance Recording for Services

- General service-oriented system (SOA)
- Provenance service that records p-assertions
  - messages send by services about their (provenance) behaviour
- Provenance store: Service that records and persists p-assertions
- Developers have three options for creating p-assertions
  - Web-service calls (SOAP, Simple Object Access Protocol)
  - Java API (Java library generates SOAP calls)
  - Instrumentation
    - Automatic generation of p-assertions by wrapping services
    - Only for services implemented using Apache Axis Web Services libraries

GY

A Graph Model of Data and Workflow Provenance

- Transformations expressed in DFL (data flow language)
- DFL is an extension of NRC (nested relational calculus)
- Define a graph model for execution of DFL expressions
- Using Haskell (functional programming language)
- Haskell implementation that generates a provenance graph as a side-effect of computation
  - Executes DFL expression
  - Generates provenance graph
  - Provenance graph is Haskell program (executable)
- Very detailed and precise (maybe too detailed?)
- Datalog queries for extracting Where- and Why-provenance from provenance graphs

Outline

Recap

Workflow Management Systems

- Design of Workflows
- Execution Engine for Workflows
- Atomic tasks
  - Services (Web-services)
  - Processes
  - ...

ILLINOIS INSTITUTE OF TECHNOLOGY

Recap

Provenance for Workflows

- Often coarse-grained
- Some fine-grained approaches
  - Kepler (UC Davis)
  - Using Pig (UPenn)
  - Provenance for Generalized MapReduce Workflows GMRW (Stanford)
- Many different models
- Standard approach
  - Workflow enactment engine serves as supervised execution environment
  - Tracks provenance during workflow execution

Recap

Covered Approaches

- Kepler (UC Davis)
- Using Pig (UPenn)
- Provenance for Generalized MapReduce Workflows GMRW (Stanford)
- The Open Provenance Model (OPM)
- Karma 3 (Indiana University)
- PReServ (University of Southampton)
- A graph model of data and workflow provenance (University of Edinburgh and others)

GY

Literature I


Sylvia C. Wong, Simon Miles, Weijian Fang, Paul Groth, and Luc Moreau.
Validation of E-Science Experiments using a Provenance-based Approach.
In AHM '05: Proceedings of the UK OST e-Science All Hands Meeting, 290-296, 2005.


Tommy Ellkvist, David Koop, Erik W. Anderson, Juliana Freire, and Claudio T. Silva.
Using Provenance to Support Real-Time Collaborative Design of Workflows.
In IPAW '08: International Provenance and Annotation Workshop, 266–279, 2008.


T. Ellkvist, L. Universitet, D. Koop, J. Freire, C. Silva, and L. Strömbäck.
Using Mediation to Achieve Provenance Interoperability.
In Proceedings of the 2009 Congress on Services-I-Volume 00, 291–298, IEEE Computer Society,2009.


Chilukuri K. Mohan.
Tutorial: State of the Art in Workflow Management System Research and Products.
5th International Conference on Extending Database Technology, Avignon, France, March, 1996.


N.N. Vijayakumar and B. Plale.
Tracking Stream Provenance in Complex Event Processing Systems for Workflow-Driven Computing.
VLDB, 2007.


S. Cohen, S. Cohen-Boulakia, and S. Davidson.
Towards a model of provenance and user views in scientific workflows.
Data Integration in the Life Sciences, page 264–279, 2006.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature II


T. Oinn, M. Greenwood, MJ Addis, M.N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, DJ Marvin, and others.
Taverna: Lessons in creating a workflow environment for the life sciences.
Journal of Concurrency and Computation: Practice and Experience, 2002.


D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M.R. Pocock, P. Li, and T. Oinn.
Taverna: A Tool for Building and Running Workflows of Services.
Nucleic acids research, 34(Web Server issue):W729, 2006.


Yong Zhao, Mihael Hategan, Ben Clifford, Ian T. Foster, Gregor Von Laszewski, Ioan Raicu, Tiberiu Stef-Praun, and Michael Wilde.
Swift: Fast, Reliable, Loosely Coupled Parallel Computation.
In IEEE Workshop on Scientific Workflows, Citeseer,2007.


A. Ailamaki, YE Ioannidis, and M. Livny.
Scientific workflow management by database management.
Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on, page 190–199, 1998.


Bertram Ludscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew B. Jones, Edward A. Lee, Jing Tao, and Yang Zhao.
Scientific Workflow Management and the Kepler System.
Concurrency and Computation: Practice and Experience, 18(10):1039–1065, 2006.


Timothy McPhillips, Shawn Bowers, Daniel Zinn, and Bertram Ludscher.
Scientific Workflow Design for Mere Mortals.
Future Generation Computer Systems, 25(5):541–551, 2008.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature III


Martin Szomszor and Luc Moreau.
Recording and Reasoning over Data Provenance in Web and Grid Services.
In ODBASE'03: International Conference on Ontologies, Databases and Applications of SEmantics, volume 2888 of Lecture Notes in Computer Science, 603–620, Catania, Sicily, Italy, November 2003.


Hyunjung Park, Robert Ikeda, and Jennifer Widom.
Ramp: a system for capturing and tracing provenance in mapreduce workflows.
In 37th International Conference on Very Large Data Bases (VLDB),Stanford InfoLab, August 2011.


Carlos Eduardo Scheidegger, Huy Vo, David Koop, Juliana Freire, and Claudio T. Silva.
Querying and Re-using Workflows with VisTrails.
In SIGMOD '08: Proceedings of the 34th SIGMOD International Conference on Management of Data, 1251–1254, ACM,2008.


O. Biton, S. Cohen-Boulakia, S.B. Davidson, and C.S. Hara.
Querying and Managing Provenance through User Views in Scientific Workflows.
Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, page 1072–1081, 2008.


Y. Amsterdamer, S.B. Davidson, D. Deutch, T. Milo, J. Stoyanovich, and V. Tannen.
Putting Lipstick on Pig: Enabling Database-style Workflow Provenance.
Proceedings of the VLDB Endowment, 5(4):346–357, 2011.


Sylvia C. Wong, Simon Miles, Weijian Fang, Paul Groth, and Luc Moreau.
Provenance-based Validation of E-Science Experiments.
In ISWC '05: Proceedings of 4th Internation Semantic Web Conference, 801–815, 2005.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature IV


Robert Ikeda, Semih Salihoglu, and Jennifer Widom.
Provenance-based refresh in data-oriented workflows.
Technical report, Stanford University, 2010.


Robert Ikeda, Semih Salihoglu, and Jennifer Widom.
Provenance-based refresh in data-oriented workflows.
In Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM '11, 1659–1668, New York, NY, USA, 2011. , ACM.


Bin Cao, Beth Plale, Girish H. Subramanian, Ed Robertson, and Yogesh L. Simmhan.
Provenance Information Model of Karma Version 3.
In SERVICES I '09: Proceedings of the Congress on Services, 348–351, 2009.


Susan B. Davidson, Sarah Cohen-Boulakia, Anat Eyal, Bertram Ludäscher, Timothy McPhillips, Shawn Bowers, and Juliana Freire.
Provenance in Scientific Workflow Systems.
IEEE Data Engineering Bulletin, 32(4):44–50, 2007.


Shawn Bowers, Timothy McPhillips, and Bertram Ludscher.
Provenance in Collection-oriented Scientific Workflows.
Concurrency and Computation: Practice and Experience, 20(5):519–529, 2008.


R. Ikeda, H. Park, and J. Widom.
Provenance for generalized map and reduce workflows.
In CIDR 2011,Stanford InfoLab, 2011.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature V


Ilkay Altintas, Oscar Barney, and Efrat Jaeger-Frank.
Provenance Collection Support in the Kepler Scientific Workflow System.
In IPAW '06: International Provenance and Annotation Workshop, 118-132, 2006.


S. Bowers, T. McPhillips, M. Wu, and B. Ludascher.
Project histories: Managing data provenance across collection-oriented scientific workflow runs.
LECTURE NOTES IN COMPUTER SCIENCE, 4544:122, 2007.


Robert D. Stevens, Alan J. Robinson, and Carole A. Goble.
myGrid: Personalised Bioinformatics on the Information Grid.
Bioinformatics, 19(90001):302–304, 2003.


C. Pautasso and T. Heinis.
Modeling and Executing Heterogeneous Grid Workflows with JOpera for Eclipse.
, .


J. Zhao, C. Goble, R. Stevens, and D. Turi.
Mining Taverna's semantic web of provenance.
Software Focus, 20(5):463–472.


Robert Ikeda, Akash Das Sarma, and Jennifer Widom.
Logical provenance in data-oriented workflows (long version).
Technical report, Stanford University, 2012.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature VI

Ilkay Altintas, C. Berkley, E. Jaeger, M. Jones, Betram Ludäscher, and S. Mock.
Kepler: An extensible system for design and execution of scientific workflows.
In SSDBM '04: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (Poster), 423–424, 2004.

YL Simmhan, B. Plale, and D. Gannon.
Karma2: Provenance Management for Data-Driven Workflows.
International Journal of Web Services Research, 5(2):1–22, 2008.

Yong Zhao, Michael Wilde, Ian T. Foster, Jens Vckler, Thomas Jordan, Elizabeth Quigg, and James Dobson.
Grid Middleware Services for Virtual Data Discovery, Composition, and Integration.
In Middleware '04: Proceedings of the 2nd Workshop on Middleware for Grid Computing, 57–62, New York, NY, USA, 2004. , ACM Press.

Manish Kumar Anand, Shawn Bowers, Timothy McPhillips, and Bertram Ludscher.
Exploring Scientific Workflow Provenance Using Hybrid Queries over Nested Data and Lineage Graphs.
In SSDBM '09: Proceedings of the 21th International Conference on Scientific and Statistical Database Management, 237–254, 2009.

Bill Howe, Peter Lawson, Renee Bellinger, Erik W. Anderson, Emanuele Santos, Juliana Freire, Carlos Eduardo Scheidegger, Antonio Baptista, and Claudio T. Silva.
End-to-End eScience: Integrating Workflow, Query, Visualization, and Provenance at an Ocean Observatory.
In eScience '08: Proceedings of the 4th IEEE International Conference on eScience, 127–134, 2008.

Literature VII


Thomas Heinis and Gustavo Alonso.
Efficient Lineage Tracking for Scientific Workflows.
In SIGMOD '08: Proceedings of the 34th SIGMOD International Conference on Management of Data, 1007–1018, 2008.


A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou, C. Pautasso, T. Heinis, R. Grnmo, H. Hoff, A.J. Berre, M. Glittum, and S. Topouzidou.
Developing scientific workflows from heterogeneous services.
ACM SIGMOD Record, 35(2):22–28, 2006.


Peng Sun, Ziyang Liu, Susan Davidson, and Yi Chen.
Detecting and resolving unsound workflow views for correct provenance analysis.
SIGMOD '09, 2009.


T. Heinis, C. Pautasso, and Gustavo Alonso.
Design and Evaluation of an Autonomic Workflow Engine.
Proc. of the 2nd International Conference on Autonomic Computing, .


Chris Wroe, Carole A. Goble, Mark Greenwood, Phillip Lord, Simon Miles, Juri Papay, Terry Payne, and Luc Moreau.
Automating experiments using semantic data on a bioinformatics grid.
IEEE Intelligent Systems, 19(1):48–55, 2004.


R.S. Barga and L.A. Digiampietri.
Automatic generation of workflow provenance.
Lecture Notes in Computer Science, 4145:1, 2006.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature VIII


Z. Bao, S.B. Davidson, S. Khanna, and S. Roy.
An optimal labeling scheme for workflow provenance using skeleton labels.
In Proceedings of the 2010 international conference on Management of data, 711–722, ACM,2010.


J. Yu and R. Buyya.
A taxonomy of scientific workflow systems for grid computing.
ACM Sigmod Record, 34(3):49, 2005.


Daniel Crawl and Ilkay Altintas.
A Provenance-Based Fault Tolerance Mechanism for Scientific Workflows.
In IPAW '08: International Provenance and Annotation Workshop, 152–159, 2008.


Yong Zhao, James E. Dobson, Ian T. Foster, Luc Moreau, and Michael Wilde.
A Notation and System for Expressing and Executing Cleanly Typed Workflows on Messy Scientific Data.
ACM Sigmod Record, 34(3):37–43, 2005.


Shawn Bowers, Timothy McPhillips, Bertram Ludscher, Sarah Cohen, and Susan Davidson.
A Model for User-Oriented Data Provenance in Pipelined Scientific Workflows.
In IPAW '06: International Provenance and Annotation Workshop, 133–147, 2006.


Umut Acar, Peter Buneman, James Cheney, Jan van den Bussche, Natalia Kwasnikowska, and Stijn Vansummeren.
A graph model of data and workflow provenance.
In TaPP '10, 2010.


Yogesh L. Simmhan, Beth Plale, Dennis Gannon, and Suresh Marru.
A framework for collecting provenance in data-centric scientific workflows.
, 2006.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature IX

S. Bowers and B. Ludascher.
A calculus for propagating semantic annotations through scientific workflow queries.
LECTURE NOTES IN COMPUTER SCIENCE, 4254:712, 2006.

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

OS Provenance Challenges

- Different Use Cases
  - Detect Errors in Execution at much lower level
  - Detect Security Breaches
- Very fine-granular and detailed provenance
  - System calls, reading/writing files/pipes
  - IPC (Inter process communication): shared memory,
  - Network communication: socket communication
  - Transformations are executable programs
  - Environment
    - Environment variables
    - Users
    - Library and OS version
    - ...

Excursion Operating Systems

- Disclaimer: A very very brief overview of some concepts
- Take of CS450, CS550, CS551 for in-depth coverage

ILLINOIS INSTITUTE OF TECHNOLOGY

Excursion Operating Systems

Virtual Memory

- Multitasking operating system
  - Multiple processes share the physical memory of a machine
  - ⇒Memory regions of these processes may be interleaved
  - ⇒Complicates application programming
- Approach: Each process has its own virtual memory address space
  - Hides fragmentation of physical memory of a process
  - ⇒Programmer does not need to worry about fragmentation
- Virtual memory can be larger than available physical memory
  - OS takes care of swapping: storing/retrieving pages from disk
    - Pages are organizational unit of memory for swapping
  - ⇒From the applications point of view memory is almost unlimited
  - ⇒No need to implement swapping in each application

Excursion Operating Systems

Processes and Threads

- Process
  - Heavy-weight construct
    - Process are associated with large state
    - Switching between processes is expensive (context-switch)
  - Usually execution of program
    - . . . but, process can start child processes
  - Each process has its own memory space
  - OS provides support for communication between processes
- Thread
  - Light-weight construct
    - Context-switch is faster than for processes
    - Less resources needed for each operating system thread state
  - Threads of the same process share memory space

Excursion Operating Systems

Inter Process Communication (IPC)

- Methods for communication between processes
- Filesystem
  - One process writes to a file and another reads from the file
- Signals
  - Stop, kill, or continue a process
  - Informs process of errors: e.g., SIGSEGV is send by POSIX OS if process accesses invalid virtual memory address
- Sockets
  - Network communication
  - Can also be used in one machine
    - E.g., DBMS server and client running on the same machine communicating via TCP

Excursion Operating Systems

IPC cont.

- Message Queues
  - Asynchronous communication:
    - One process submits messages to the queue
    - Other process reads message from queue
- Pipes
  - Connects processes stdin and stdout
- Semaphores
  - Control access to limited resources by multiple processes
- Shared memory
  - Memory that can be accessed by more than one process

Excursion Operating Systems

File-systems (UNIX)

- Organize content on disk
- Some linux supported file-systems: ext2/3/4, ReiserFS
- Store file data and meta-data
  - e.g., creation-time
- Meta-data stored in inodes

ILLINOIS INSTITUTE OF TECHNOLOGY

Excursion Operating Systems

Kernel vs. User space

- Kernel monopolizes (abstracts from) access to hardware
  - Memory
  - File-systems
  - Graphics
- Kernel realizes scheduling of processes
- Kernel space vs. user space
  - Applications (user space) access hardware indirectly through kernel interface (system calls)

ILLINOIS INSTITUTE OF TECHNOLOGY

Approaches

- Provenance-aware file systems
  - Lineage File System
  - PASS
  - Praana: A Personalized Desktop Filesystem
- Kernel-extensions (instrumented kernels)
  - Hi-Fi
  - PASS
- Provenance support in virtualization layers
  - Provenance support via the Xen Hypervisor
- Instrumenting Executable Programs to track provenance
  - Extend Valgrind for provenance collection

Outline

PASS

- File- and process-level provenance
- Modified Linux Kernel
  - Provenance capturing is automatic and transparent to applications
- Provenance model
  - Graph
  - Nodes are files and processes (and sockets)
  - Edges between processes and files indicate
    - Process reads file
    - Process writes to file

ILLINOIS INSTITUTE OF TECHNOLOGY

PASS

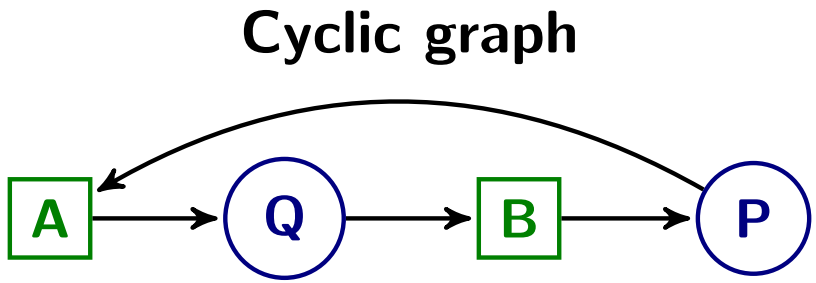
Architecture

- Pasta: The provenance and storage layer
  - File-system that tracks provenance
  - Old-versions are not persisted ⇒no versioning
  - Provenance records (p-nodes) are never deleted
  - KBDB: Kernel implementation of Berkley DB
    - stores provenance information
- Provenance Collector
  - Provenance Data Recording
  - Breaking Cycles
    - Avoid generating a new object version on each change
    - Merging nodes during cross-referencing
- Query tool

PASS

Example

	Process P	Process Q
1		read A
2		write B
3	read B	
4	write A	



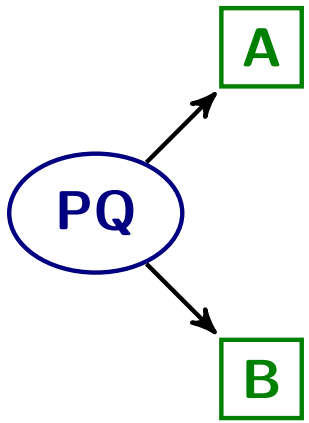
ILLINOIS INSTITUTE OF TECHNOLOGY

PASS

Example

	Process P	Process Q
1		read A
2		write B
3	read B	
4	write A	

After Merging



ILLINOIS INSTITUTE OF TECHNOLOGY

Lineage File System

- Modified Linux kernel
- System calls for process creation and file manipulation instrumented
  - create provenance records in kernel buffer (`printk`)
- User daemon collects provenance
  - Wakes up regularly
  - Writes provenance to MySQL database

ILLINOIS INSTITUTE OF TECHNOLOGY

Praana: A Personalized Desktop Filesystem

- File-system with support for annotations
  - extends ext3
- Metadata Repository
  - Relational Database
  - Relation Graphs
    - Each file is a node
    - Edges between files in same directory
    - New edges based on provenance
- Inference Learning Engine (ILE)
  - "Learns" edge weights in graph based on user feedback and provenance

OF TECHNOLOGY

Provenance using Xen Hypervisor

- Xen Hypervisor
  - Virtual machine engine
- Adapt the PASS approach for instrumenting the Xen Hypervisor
- $\Rightarrow$Enables provenance collection for virtual machines

ILLINOIS INSTITUTE OF TECHNOLOGY

Instrumenting Programs using Valgrind

- Use program instrumentation framework Valgrind
- Instrument programs using dataflow analysis
- Dynamic dataflow analysis
  - For a variable and a given part of a program
  - Determine which variable assignments (memory values) influence the variable at this position
  - Do that for a given input (execution)
- Use slicing to determine which input reads influence an output statement (write)
- $\Rightarrow$get very fine-grained provenance for arbitrary executable programs

Instrumenting Programs using Valgrind

Valgrind

- Program instrumentation framework
- Used mainly for debugging memory errors
  - detects access beyond array boundaries
  - detects access to previously free'd memory
  - detects memory holes: memory that is not referenced from anywhere but was not free'd
  - ...
- Functionality is achieved by instrumenting programs
  - $\Rightarrow$Modify the program's (machine) code

ILLINOIS INSTITUTE OF TECHNOLOGY

Instrumenting Programs using Valgrind

Dataflow Analysis

- Track on which variables the value of a variable at a certain point in the program depends on
- Using dependency relations
  - $DEP(x, i)$: variables the value of variable x at line i depends on
- Computed recursively for static analysis
- Computed by forward construction for dynamic analysis

ILLINOIS INSTITUTE OF TECHNOLOGY

Instrumenting Programs using Valgrind

Example

```
 1  read(x,y);
 2  total = 0;
 3  sum = 0
 4  if (x <=1) {
 5     sum = y;
 6  }
 7  else {
 8     read(z);
 9     total = x * y * z;
10  }
11  write(total, sum);
```

Example

Input: $x = 1,\ y = 5$

$DEP(sum, 11) = \{y\}$

Input: $x = 2,\ y = 3,\ z = 2$

$DEP(total, 11) = \{x, y, z\}$

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

Recap

OS Provenance Challenges

- Different use cases that need very fine-granular provenance
  - Detecting Security Breaches
  - Detecting Execution Anomalies
  - Repeatability
  - Auditing
- Very low-level provenance
- Many approaches work for arbitrary programs

ILLINOIS INSTITUTE OF TECHNOLOGY

Recap

Covered Approaches

- PASS
- Lineage File System
- Praana
- Xen Hypervisor provenance
- Provenance using program instrumentation with Valgrind

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature I


B. Korel and J. Laski.
Dynamic program slicing.
Information Processing Letters, 29(3):155–163, 1988.


Bogdan Korel and Satish Yalamanchili.
Forward Computation of Dynamic Program Slices.
In ISSTA '94: Proceedings of the International Symposium on Software Testing and Analysis, 66–79, 1994.


Bogdan Korel and Jurgen Rilling.
Dynamic Program Slicing Methods.
Information and Software Technology, 40(11-12):647–659, 1998.


M.I. Seltzer, P. Macko, and M.A. Chiarini.
Collecting provenance via the xen hypervisor.
In Proceedings of 3rd USENIX Workshop on the Theory and Practice of Provenance (TaPP'11), 2011.


Mingwu Zhang, Xiangyu Zhang, Xiang Zhang, and Sunil Prabhakar.
Tracing Lineage beyond Relational Operators.
In VLDB '07: Proceedings of the 33th International Conference on Very Large Data Bases, 1116–1127, VLDB Endowment, , 2007.


A. Roy and S. Balasubramaniam.
Praana: a personalized desktop filesystem.
, 2010.


Can Sar and Pei Cao.
Lineage file system.
Technical report, Stanford University, 2005.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature II


Margo Seltzer, Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Jonathan Ledlie.
Provenance-Aware Storage Systems.
Technical report, Harvard University, 2005.


Jonathan Ledlie, Chaki Ng, David A. Holland, Kiran-Kumar Muniswamy-Reddy, Uri Braun, and Margo Seltzer.
Provenance-Aware Sensor Data Storage.
In ICDE '03: Workshop Proceedings of the 21th International Conference on Data Engineering, 1189, , , 2005.


K.K. Muniswamy-Reddy, P. Macko, and M. Seltzer.
Provenance for the cloud.
In Proceedings of the 8th USENIX conference on File and storage technologies, 15–14, USENIX Association, , 2010.

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

Distributed Systems

- Disclaimer: Very limited discussion needed to understand the presented approaches
- Take classes for in-depth discussion: one of the many distributed systems classes including cloud computing ...
- HPC-clusters or Grids:
  - Distributed processing of expensive computations
    - e.g., physics simulations
  - on a large number of machines
- Overlap with workflow systems:
  - many of the discussed workflow systems are actually distributed systems

GY

Scripting Languages for Distributed Computation

- Makes distribution to some degree transparent
  - Translates script into a bunch of jobs
  - These jobs are executed distributively using a job submission system
  - Determine when to move load and/or data from one host to another
  - Failover
- Chimera: VDL
- Swift
- JSDL
- ...

Outline

Chimera

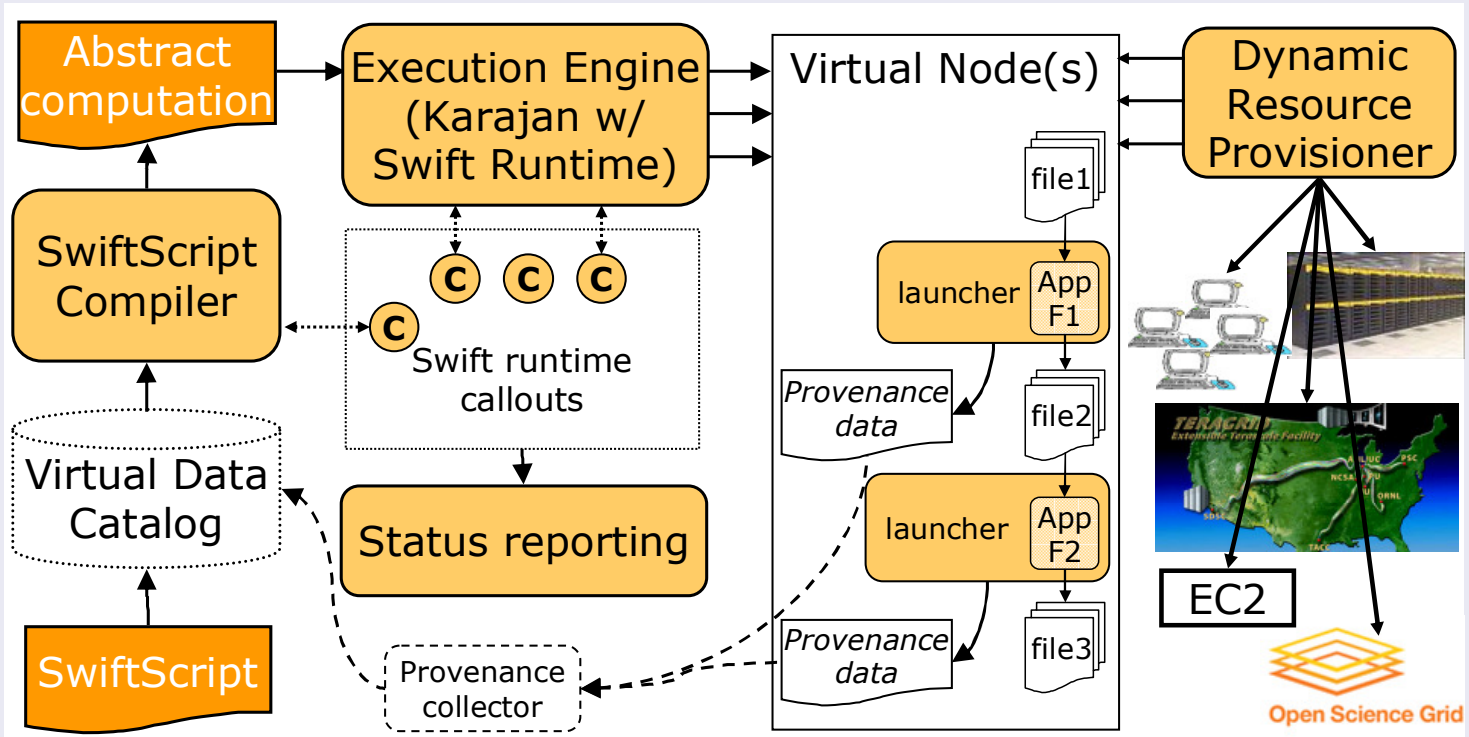
- VDL (Virtual Data Langage)
  - Describe data set properties
  - "Script" large-scale distributed executions
- Catalog that stores
  - Transformation descriptions
  - Execution descriptions
  - Data set descriptions
  - Provenance

ILLINOIS INSTITUTE OF TECHNOLOGY

Swift

- Extension of Chimera
- Concise specification and efficient execution of large loosely coupled computations
- Scripting Langauge **SwiftScript**
  - Describes Data
    - **logical** and **physical** structure of the dataset to be processed
    - Mappers to map data from and to XML
  - Task to execute
- **Karajan** and **Falkon**: job submission system and compiler
- Provenance generator: **Kickstart**
  - wraps the executables
  - tracks their behaviour

Architecture



ILLINOIS INSTITUTE OF TECHNOLOGY

Tracking and Sketching Distributed Provenance

- Extends ideas of PASS and other OS approaches
- For distributed computations
- Provenance model includes location and references to provenance on other hosts
- Less centralized storage model
  - each host stores its own provenance
- Track network traffic to detect inter-host dependencies
  - e.g., ssh
- Provenance Sketches
  - Compressed approximate representation of provenance
  - using bloom filters

Outline

Recap

Provenance for distributed systems

- Scripting languages for distributed computation
  - Chimera
  - Swift
  - ...
- Tracking and Sketching Distributed Data Provenance
  - Similar to OS-based approaches
  - but tailored for distribution
    - by considering inter-host dependencies
    - distributed storage of provenance
  - Provenance Sketches
    - compact approximate representation of provenance
    - uses Bloom filters

Literature I


Ian T. Foster, Jens-S. Vckler, Michael Wilde, and Yong Zhao.
Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation.
In SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management, 37–46, 2002.


James Annis, Yong Zhao, Jens Voeckler, Michael Wilde, Steve Kent, and Ian T. Foster.
Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey.
In Supercomputing '02: Proceedings of the Conference on Supercomputing, 1–14, 2002.


Yong Zhao, Michael Wilde, and Ian T. Foster.
Applying the Virtual Data Provenance Model.
In IPAW '06: International Provenance and Annotation Workshop, 148–161, 2006.


Yong Zhao, Mihael Hategan, Ben Clifford, Ian T. Foster, Gregor Von Laszewski, Ioan Raicu, Tiberiu Stef-Praun, and Michael Wilde.
Swift: Fast, Reliable, Loosely Coupled Parallel Computation.
In IEEE Workshop on Scientific Workflows, Citeseer, , 2007.


Ben Clifford, Ian T. Foster, Jens S. Voeckler, Michael Wilde, and Yong Zhao.
Tracking Provenance in a Virtual Data Grid.
Concurrency and Computation: Practice and Experience, 20(5):565-575, 2008.


T. Malik, L. Nistor, and A. Gehani.
Tracking and sketching distributed data provenance.
In e-Science (e-Science), 2010 IEEE Sixth International Conference on, 190–197, IEEE, , 2010.

ILLINOIS INSTITUTE OF TECHNOLOGY