Accelerating Scientific Discovery by Tracking Data Provenance

Presenter: Dong Dai, DISCL Lab. TTU

A work done with Dr. Yong Chen (TTU), Dr. Robert Ross (ANL), Dr. Phil Carns (ANL), and Dr. John Jenkins (ANL)
Outline

• Provenance: A Quick Introduction
• Motivation: A Scientific Example
• LPS: Lightweight Provenance Service
• LPS: Performance Evaluation
• Conclusion & Current Work
Provenance
Provenance: A Quick Introduction

• **Provenance**, also referred as lineage, is metadata that describe the history of a piece of data [In Computer Science]

• It contains:
  • Links between results and corresponding starting conditions or configuration parameters
  • Links between a data piece and applications
  • and many other attributes information...

• It provides the ability for users to point at a piece of data and ask: **where did it come from?**
Provenance: A Quick Introduction

- OPM (Open Provenance Model) is the official model for provenance data.
- Provenance is normally organized as graph with properties
- Queries on provenance normally are translated to graph operations

http://www.b-me.de/?page_id=8
W3C Provenance Incubator Group Use Cases

- Result Differences
- Anonymous Information
- Information Quality Assessment for Linked Data
- Timeliness
- Simple Trustworthiness Assessment
- Ignoring Unreliable Data
- Answering user queries that require semantically annotated provenance
- Provenance in Biomedicine
- Closure of Experimental Metadata
- Provenance Tracking in the Blogosphere
- Provenance of a Tweet
- Provenance and Private Data Use
- Provenance of Decision Making in Emergency Response
- Provenance of Collections vs Objects in Cultural Heritage
- Provenance at different levels in Cultural Heritage
- Locating Biospecimens With Sufficient Quality
- Identifying attribution and associations
- **Determining Compliance with a License**
- Documenting axiom formulation
- Evidence for public policy
- Evidence for engineering design
- Fulfilling Contractual Obligations
- Attribution for a versioned document
- **Provenance for Environmental Marine Data**
- Crosswalk Maintenance
- Metadata Merging
- Mapping Digital Rights
- Computer Assisted Research
- Handling Scientific Measurement Anomaly
- Human-Executed Processes
- Semantic disambiguation of data provider identity
- Hidden Bug
- Using process provenance for assessing the quality of Information products

http://www.w3.org/2005/Incubator/prov/wiki/Use_Cases
An Example of Provenance: Blog Aggregators

- It wants to determine the correct originator of an item
- It wants to determine if it can reuse an image
- It wants to determine if the image was modified
- It wants to provide confidence to the end-user

http://www.w3.org/2005/Incubator/prov/wiki/Use_Cases
An Example of Provenance: Blog Aggregators

http://www.w3.org/2005/Incubator/prov/wiki/Use_Cases
Motivation: Scientific Cases
Motivation: A Scientific Example

• Assume a workflow\textsuperscript{1} for creating population-based "brain atlases" from the high resolution fMRI anatomical data.

1. The inputs are a set of new brain images and a single reference brain image.
2. For each image, there is the actual image and the metadata information for that image
3. The output is a graphical atlas image converted from series of processing

\textsuperscript{1} First Provenance Challenge, http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge
The orange ones are procedures

The blue ones are data items

First Provenance Challenge,
http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge
Motivation: A Scientific Example

• **Understanding how a result is generated!**

• Assume we have thousands of images that have been processed through this workflow (including intermediate results).

  1. Find all the processes that led to a given Atlas X Graphic.
  2. Find all invocations of procedure `align_warp` using parameter `-m 12` that ran on a Monday.
  3. Find all graphic images outputted from workflows where at least one of the input Anatomy Headers had an entry `maximum=4095`.

Provenance gives answers for these questions!
LPS: Lightweight Provenance System for HPC Platforms
Existing Provenance Systems

- Provenance systems have been widely researched.

However, there is not a provenance system for HPC yet.

Figure from “Lucian, et. al. A Primer on Provenance, Communication of ACM, May, 2014”
Existing Provenance Systems - Problems

- **Performance Consideration:**
  - Collecting provenance introduces overhead, however, HPC is performance sensitive.
  - Expect less than 1% slowdown and a small memory footprint on any provenance system running in HPC.

- **Granularity Requirements**:
  - HPC applications could be workflow, parallel job, or even single process.
  - Data could be shared and accessed concurrently.
  - Finer granularity provenance like process accessed a file on certain time is needed.

- **Simplicity Consideration**:
  - No extra modification on applications or users' behaviors should be allowed.

Existing provenance systems normally introduce more than 10% overheads.

Workflow-based provenance systems provide less 10% overheads, but failed to provide certain details.

Library-based instrument is not OK.
LPS: Lightweight Provenance System (Architecture)

- **Login Nodes**
- **Compute Nodes**
- **Users Allocated Compute Nodes**
  - Local LPS Collector
  - Local LPS Aggregator
- **LPS Dedicated Nodes**
  - Distributed LPS Aggregator
- **Data**
- **Provenance**
- **Parallel File System**
LPS: Local Provenance Collector

- **Operation System Instrumentation**
  - Systemtap script
  - Probe on critical system calls, like `create/exit`, `open/close`, `read/write`, etc.
  - Read `/proc` file system

- **Dynamic Instrumentation**
  - Two systemtap scripts, one only captures basic system calls like `open/close`; one captures frequent ones like `read/write`
  - Dynamic adjust by end users

- **Advantage:**
  - Transparent to users and applications
  - Covering finer granularity activities on advanced capture
  - Low overhead on basic capturing
LPS: Execution Provenance Aggregator

• Execution provenance aggregator combines processes and jobs collected in different locations together
  • In login nodes, LPS collects *job submitting* operations
  • In compute nodes, LPS collects *process executions*

• LPS uses environment variables of these processes to map entities together
  • Each MPI process will have a *PBS_JOBID* env. Variable
  • Each *job submission* command will get the same job Id as return value

• Advantage:
  • A unified interface. Any distributed computing framework, like Hadoop MapReduce, can be supported.
LPS: Data Access Provenance Aggregator

- It determines the correct data dependencies
  - Different processes of a job may access the same file
  - Different jobs may access the same file
  - Different users may access the same file
- Their order (causality) is critical as they determine the data dependencies.

- There are three different granularities on data dependency in LPS

**TABLE I: Three granularity choices in LPS.**

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Overhead</th>
<th>Compensiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open/Close*</td>
<td>Low</td>
<td>Only open/close a file</td>
</tr>
<tr>
<td>First/Last IO</td>
<td>Medium</td>
<td>Accurate IO time, Know IO Size</td>
</tr>
<tr>
<td>Interleaved IO</td>
<td>High</td>
<td>Each read/write dependency</td>
</tr>
</tbody>
</table>
LPS: Performance Evaluation
LPS: Overhead On HPCG Benchmark

Fig. 7: LPS Performance Analysis with the HPCG Benchmark.
Fig. 8: LPS Performance Analysis with the IOR Benchmark.
Fig. 9: LPS Performance Analysis with the MDTest Benchmark.
LPS: Overhead on MDTest Benchmark

Fig. 9: LPS Performance Analysis with the MDTest Benchmark.
Fig. 13: LPS distributed aggregator scalability and overhead analysis.
Conclusion & Current Work
Conclusion & Concurrent Work

• **Conclusion:**
  • It is possible to transparently collect provenance in HPC environment.
  • The overhead can be controlled under 1% if willing to loss accuracy on data dependency.
  • Dynamic changing granularity is possible in HPC environment to satisfy different users.

• **Current and Future Work:**
  • Prototype Implemented
  • Provenance query tool is undergoing
  • Deploy in production HPC cluster is planned
Thanks & Questions
#!/bin/sh
AIR5.2.5/bin/align_warp anatomy1.img reference.img warp1.warp -m 12 -q
AIR5.2.5/bin/align_warp anatomy2.img reference.img warp2.warp -m 12 -q
AIR5.2.5/bin/align_warp anatomy3.img reference.img warp3.warp -m 12 -q
AIR5.2.5/bin/align_warp anatomy4.img reference.img warp4.warp -m 12 -q
AIR5.2.5/bin/reslice warp1.warp resliced1
AIR5.2.5/bin/reslice warp2.warp resliced2
AIR5.2.5/bin/reslice warp3.warp resliced3
AIR5.2.5/bin/reslice warp4.warp resliced4
AIR5.2.5/bin/softmean atlas.hdr y null resliced1.img resliced2.img resliced3.img resliced4.img
fsl/bin/slicer atlas.hdr -x .5 atlas-x.pgm
fsl/bin/slicer atlas.hdr -y .5 atlas-y.pgm
fsl/bin/slicer atlas.hdr -z .5 atlas-z.pgm
convert atlas-x.pgm atlas-x.gif
convert atlas-y.pgm atlas-y.gif
convert atlas-z.pgm atlas-z.gif