

Performance Evaluation of the Karma Provenance Framework for Scientific Workflows

Yogesh L. Simmhan

Beth Plale, Dennis Gannon, Suresh Marru

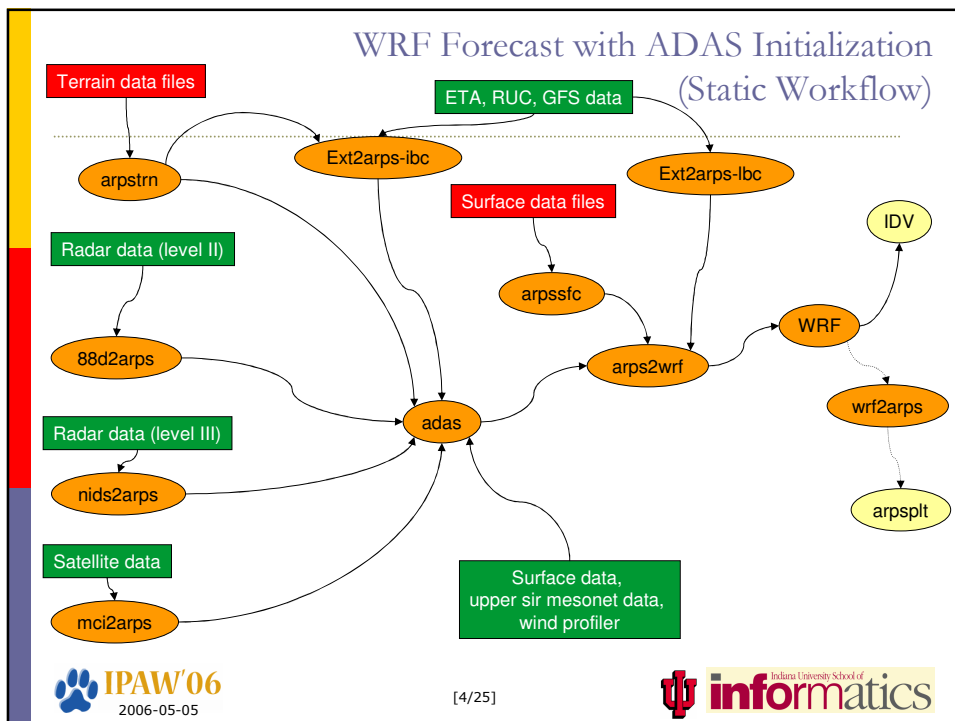
Indiana University

Outline

- Motivation
- Architecture of Karma
- Experimental Setup
- Collecting & Querying Provenance
- Analysis & Conclusion

Dynamic & Adaptive Workflows

- Linked Environments for Atmospheric Discovery (LEAD) project
- Weather & Severe Storm Prediction Applications
- **Dynamic:** Structure & Semantics of workflow may change due to external events. E.g. detection of a hurricane during run may trigger another workflow, or steer current workflow's direction
- **Adaptive:** How & where the workflow executes is determined at runtime



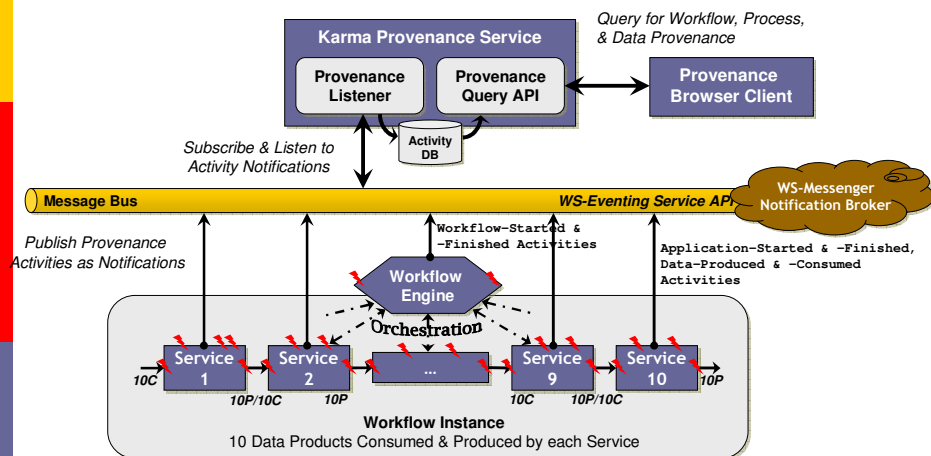
Provenance Collection: Challenges & Uses

- Provenance on workflow (process) & data products at fine granularity
- Dynamic, Long running workflows
- Helps scientists to
 - Select workflows & data products
 - Build workflows
 - Track workflow execution
 - Analyze data products from runs

Karma Provenance Framework

- **Lightweight** – do not duplicate available metadata cataloging effort
 - myLEAD personal metadata catalog
 - ResCat service & data registry
- Glue to integrate metadata on data & services with runtime workflow information
- **Scalability** – 500 users, 100's of workflows, 10,000's of data products

Karma Architecture²

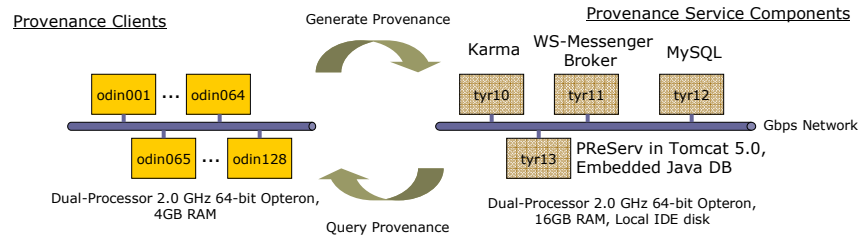


[2] *A Framework for Collecting Provenance in Data-Centric Scientific Workflows*, Simmhan, Y., et al., Submitted to ICWS Conference, 2006

Provenance Performance Study

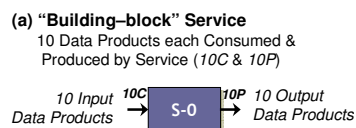
- Come up with workloads to benchmark scalability of provenance system
- Verify scalability of Karma with these workflow & data loads
- Do baseline comparison with another contemporary provenance system: PReServ
- Monopolize the new compute cluster at our department ☺

Experimental Setup

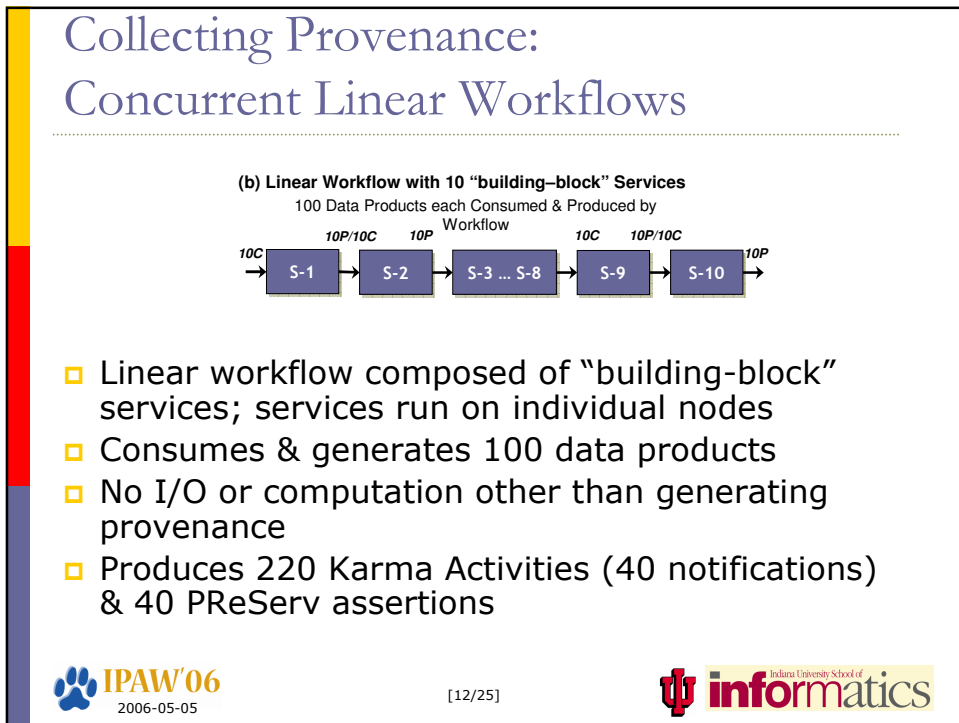
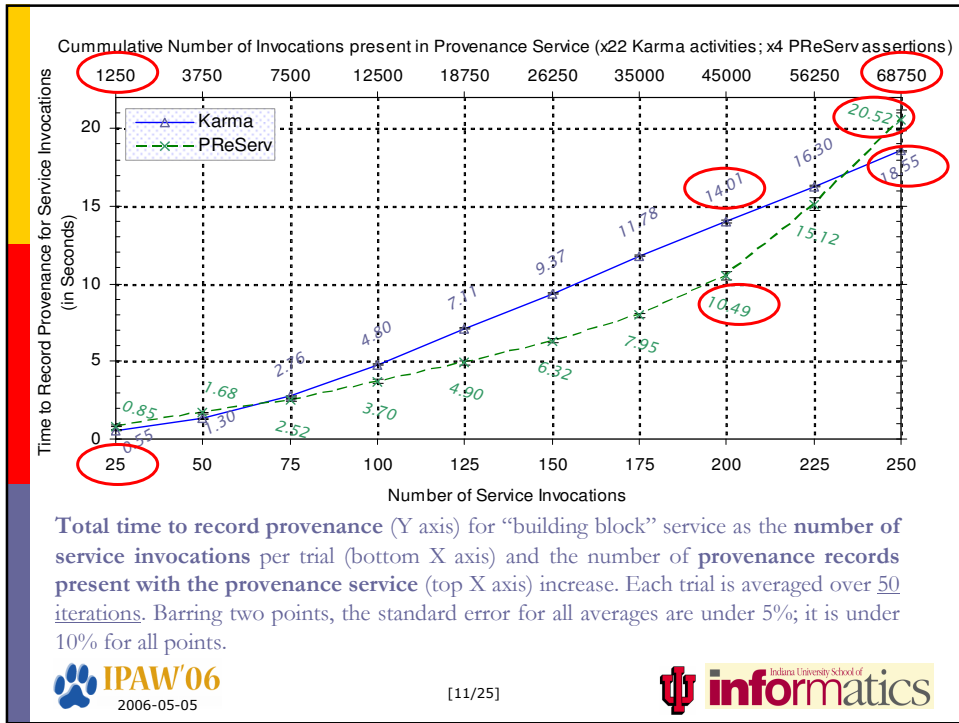


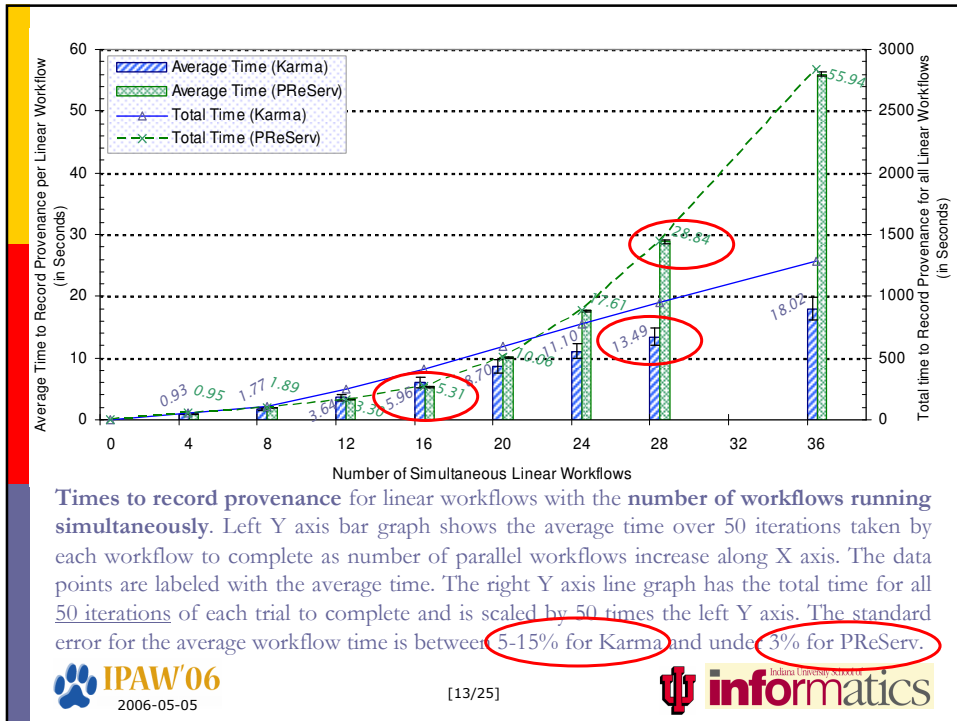
- Karma Service, WS-Messenger Notification Broker, MySQL
- PReServ in Tomcat 5.0 container
- Tyr web-services cluster (16 Nodes)
- Odin computer cluster (128 Nodes)
- Gigabit Ethernet, local IDE disk storage
- SLURM job manager for parallel job submission on *Odin*
- Java 1.5, Jython

Collecting Provenance: Simple “Building-block” Workflow

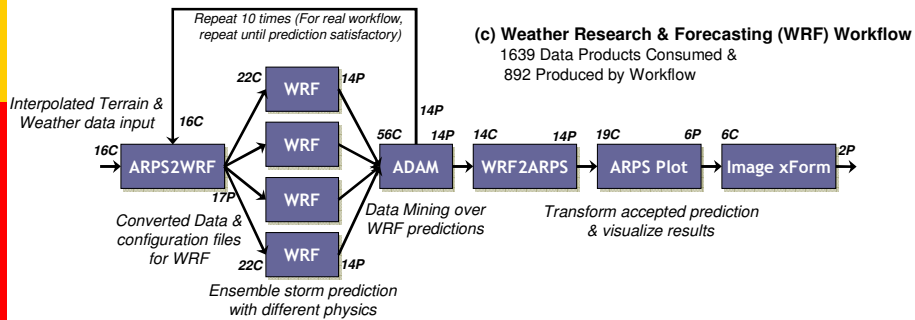


- Single Service Workflow
- Simulates consumption of 10 data products as input & generates 10 data products as output
- No I/O or computation other than generating provenance
- Produces 22 Karma Activities (4 notifications) & four PReServ assertions using *Notifier* Library

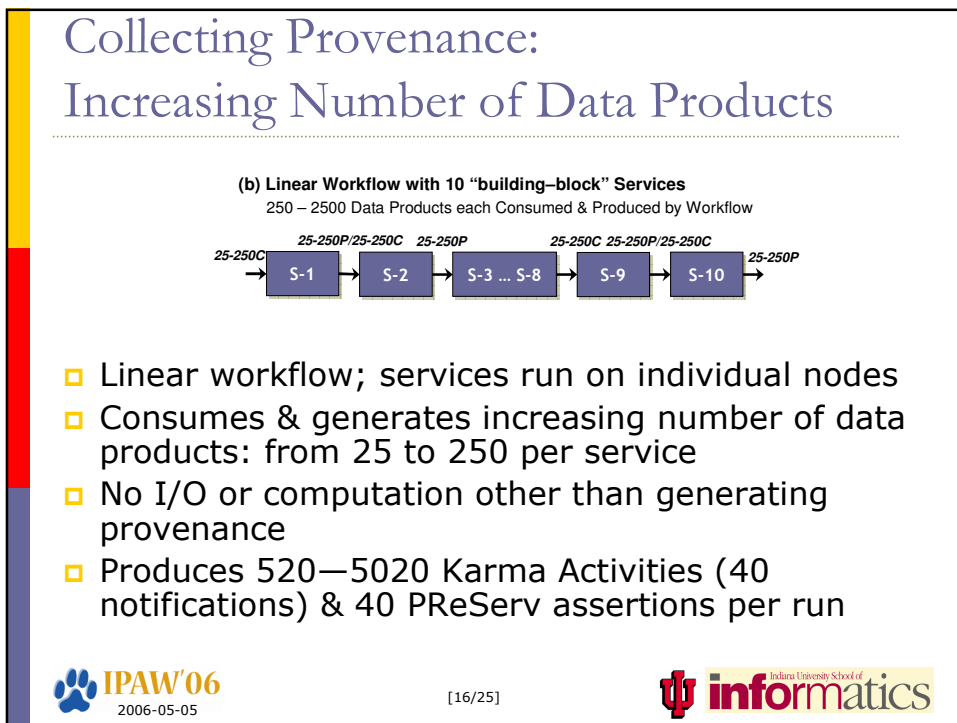
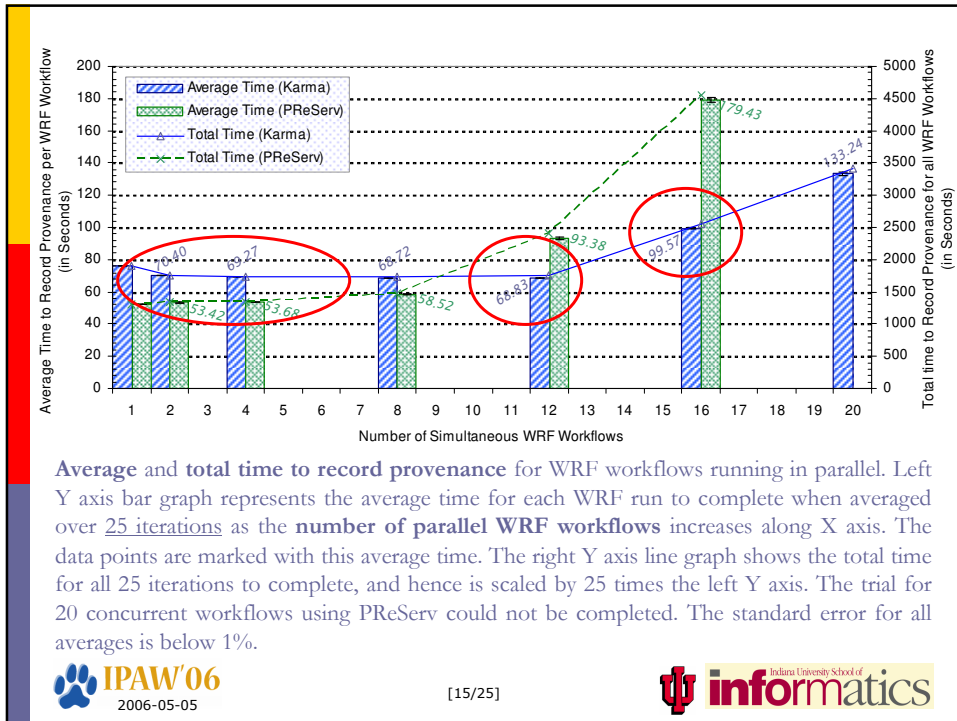


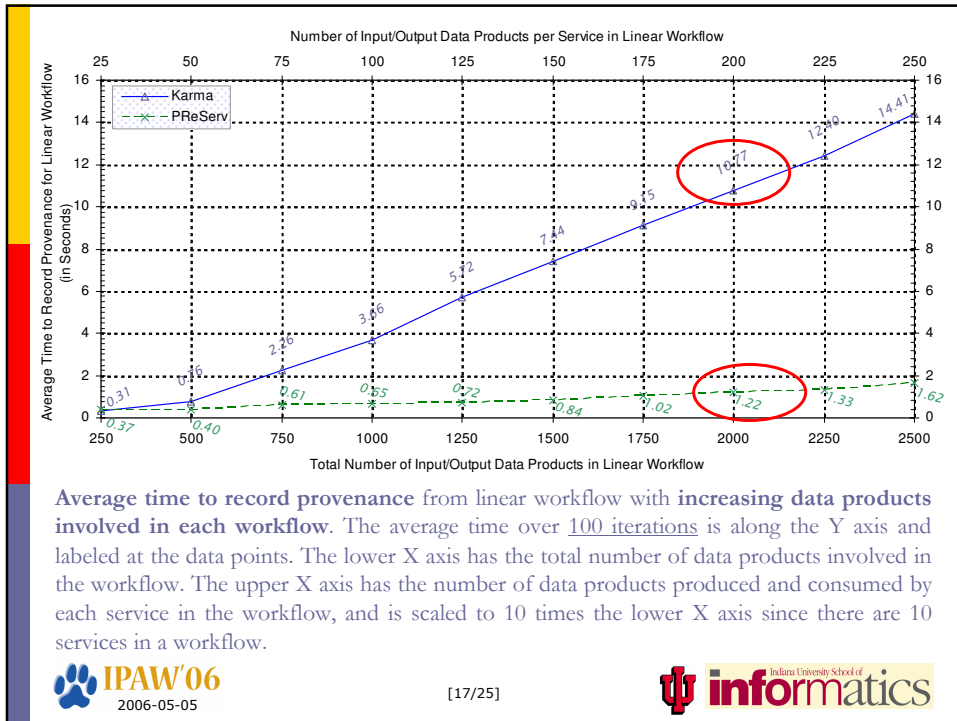


Collecting Provenance: Real-world synthetic Workflow



- WRF Forecast Simulation Workflow, identical to real-world
- Ensemble runs with iteration; services run on individual nodes
- Consumes 1639 & generates 892 data products
- No I/O or computation other than generating provenance
- Produces 2657 Karma Activities (252 notifications) & 252 PReServ assertions per run



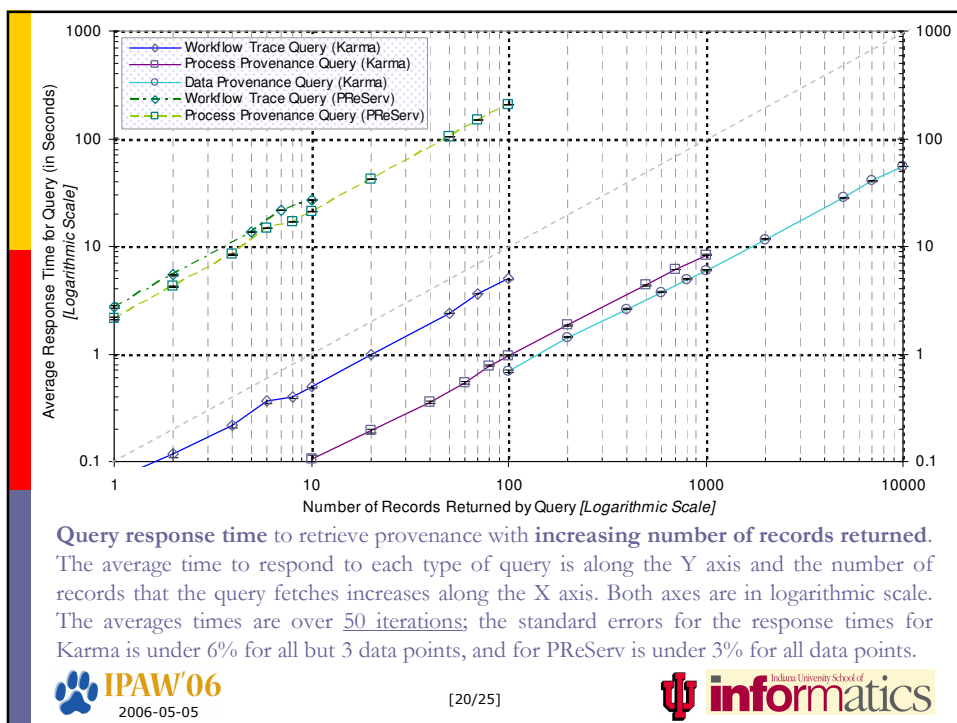


Querying Provenance

- Three types of provenance can be queried
- Process provenance
 - When was a service invoked & in which WF?
 - What were its input and output data products
- Data Provenance
 - What service & WF generated this data product & when? Which services & WFs use this & when?
- Workflow Trace
 - What were all the services invoked in this workflow & when? What data were consumed & produced by them?

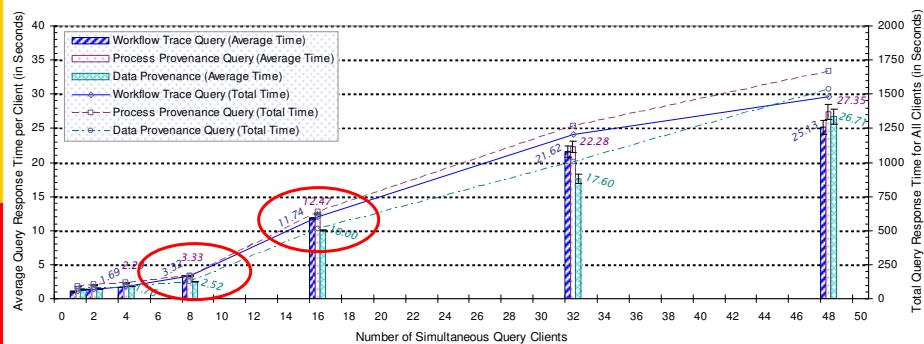
Querying Provenance: Size of Result Set

- Response time for fetching increasing number of provenance records through single query client
- Process, data & workflow provenance for Karma using WF, Service, Data ID
- Workflow & data provenance for PReServ using XPath
- Provenance Services loaded with:
 - 1000 linear WFs for Karma (10,000 service invocations; 100,000 unique data products; 220,000 activities)
 - 100 linear WFs for PReServ (1000 service invocations; 10,000 unique data products; 4000 assertions)
- Returns 0.01% – 10% of provenance documents



Querying Provenance: Number of Concurrent Clients

- Response time for multiple simultaneous clients to fetch provenance records
- 200 Process, 200 data & 20 workflow provenance documents for Karma
- *Not performed for PReServ*
- Services loaded with:
 - 1000 linear WFs for Karma (10,000 service invocations; 220,000 activities)



Query response time to retrieve 20 workflow trace, 200 process provenance, and 200 data provenance from increasing number of concurrent query clients. The average query time over 50 iterations is shown on the left Y axis bar graph, the data points being labeled with this value. The total time to complete the 50 iterations is the line graph on the right Y axis, and is accordingly scaled by 50 times compared to the left Y axis. The X axis shows the increasing number of parallel query clients. The standard errors for all averages are under 4%.

Discussion

- Karma scales well (linearly or better) for all workloads considered – for collecting & querying for provenance
- Well defined, lightweight provenance: Glue
- Use of mature software stack: MySQL, WS-Messenger, SOAP libraries
- PReServ provides flexibility & more generic provenance scheme
 - Loses out on performance; partly due to storage layer

Future Work

- Running non-synthetic workloads
- Logging of SOAP messages to rerun Workflow; Recording configuration parameters
- Ways to use provenance
 - Quality model for data products based on their provenance³
- Kinds of queries that can be done on provenance: work-patterns
- Standard ways to collect provenance

[3] [Towards a Quality Model for Effective Data Selection in Collaboratories](#), Simmhan, Y.L., et al., SciFlow Workshop, 2006

Acknowledgements

Paul Groth, Univ. of Southampton
Systems Staff, CS Department, Indiana Univ.
Ramya & Abhijit, Indiana Univ.

LEAD Members



NSF



Questions

www.extreme.indiana.edu/karma