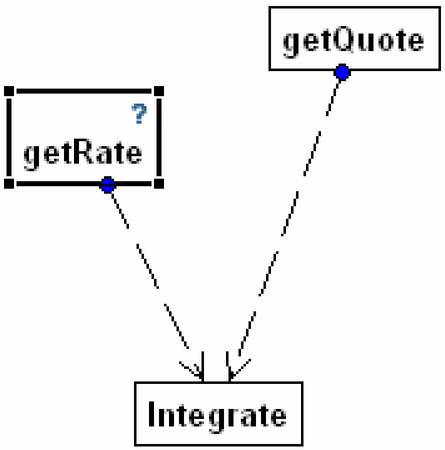
Publishing Persistent Grid Computations as WS Resources

Thomas Heinis, Cesare Pautasso, Oliver Deak, Gustavo Alonso

{heinist, pautasso,alonso}@inf.ethz.ch odeak@student.ethz.ch

Department of Computer Science, ETH Zurich

Motivation



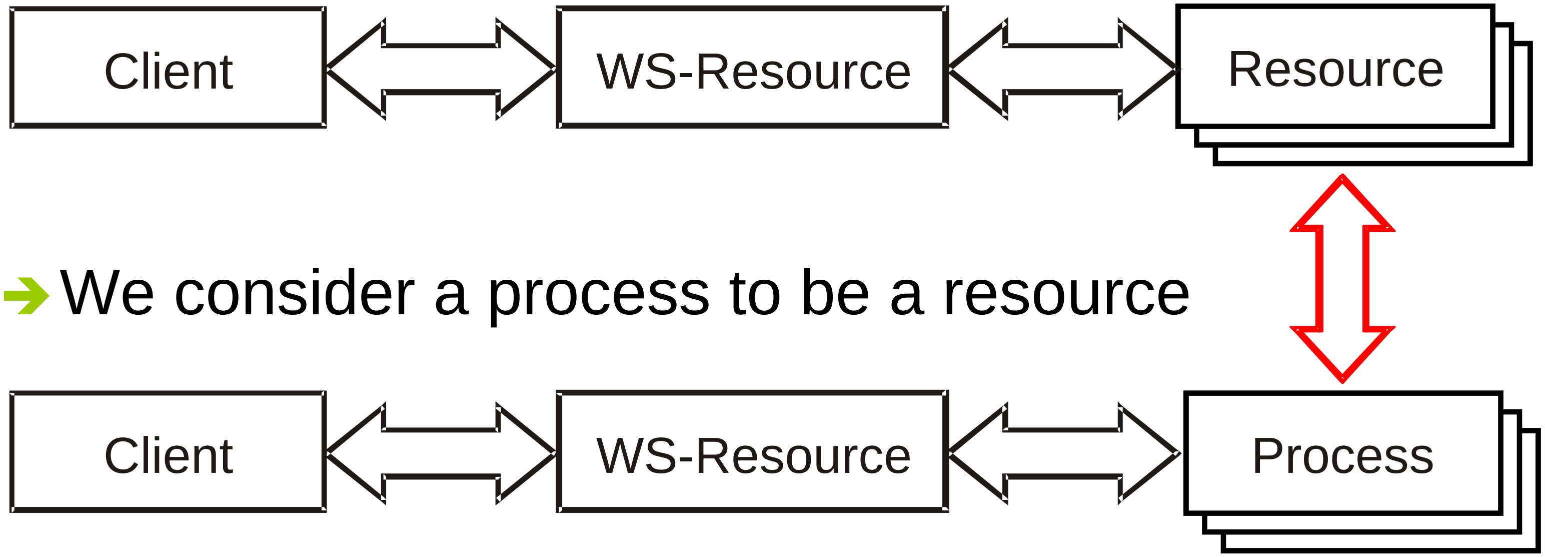
- Grid computations/workflows can be modeled as processes
- Composed processes can be published as Grid services
- Provides processes with a well-understood and standardized interface
- Eases reusability and integration (Grid clients and portals)
- To do so, we map a process to a Grid service (compliant with WS-RF and WS-N)

Interface Types

- Lifecycle Management
  - Start / suspend / resume / stop processes, discard accumulated state
- Monitoring and Steering
  - Monitoring of the process execution using polling and subscribe – notify interaction patterns
  - Steering of the process execution by suspension and change of the execution state
- Managing process batches

Mapping Processes to Grid Services

- WS-Resource specification:
  - defines the implied resource pattern



Client ⇔ WS-Resource ⇔ Resource

→ We consider a process to be a resource

Client ⇔ WS-Resource ⇔ Process

Lifecycle Management

- WS-Lifetime defines resources to have a lifetime, and defines operations to end it immediately or scheduled
- Mapping implies a process instance to have a lifetime
- No operation specified to create a resource / start a process
- We additionally defined operations to start (create resources), suspend and resume processes:
  - startProcess, suspend, resume, startSuspended

Monitoring & Steering

- WS-Properties defines resources to have properties which can be read and written
- All elements of the process execution state are considered to be properties
- Allows for synchronous interaction pattern to read and write elements of the state
- Breakpoints are used to suspend the execution

Monitoring & Steering

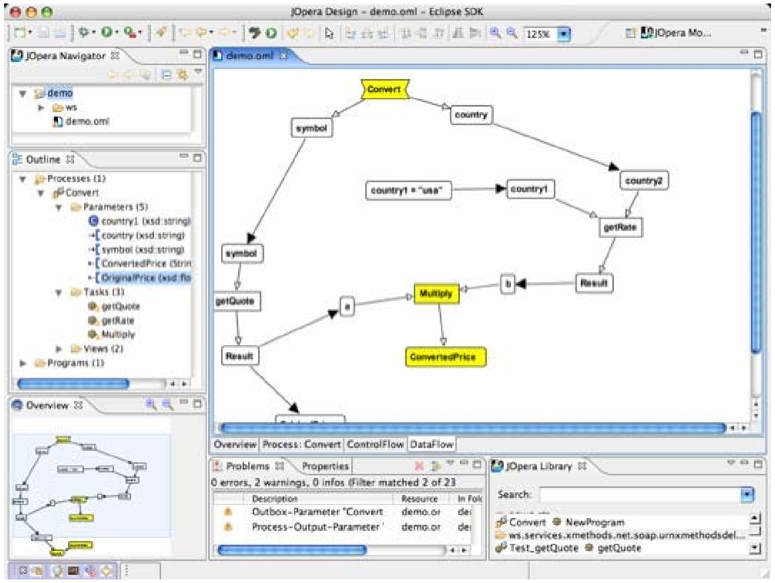
- WS-Topics defines resources to provide topics to which clients can subscribe
- WS-Notification defines operations to subscribe to topics
- All elements of the process execution state are topics
- We additionally define an operation to subscribe to a topic before starting the execution called startSubscribed

Batch Process Management

- WS-ServiceGroup defines a mechanism to classify resources
- Eases discovery and management of resources
- Process instances belonging to the same batch are grouped using service groups
- We additionally define an operation to start a batch process called startBatch

Design

- Generic approach applicable to many different process execution engines
- We have implemented the mapping in JOpera in order to show feasibility



0 errors, 2 warnings, 0 infos (Filter matched 2 of 23

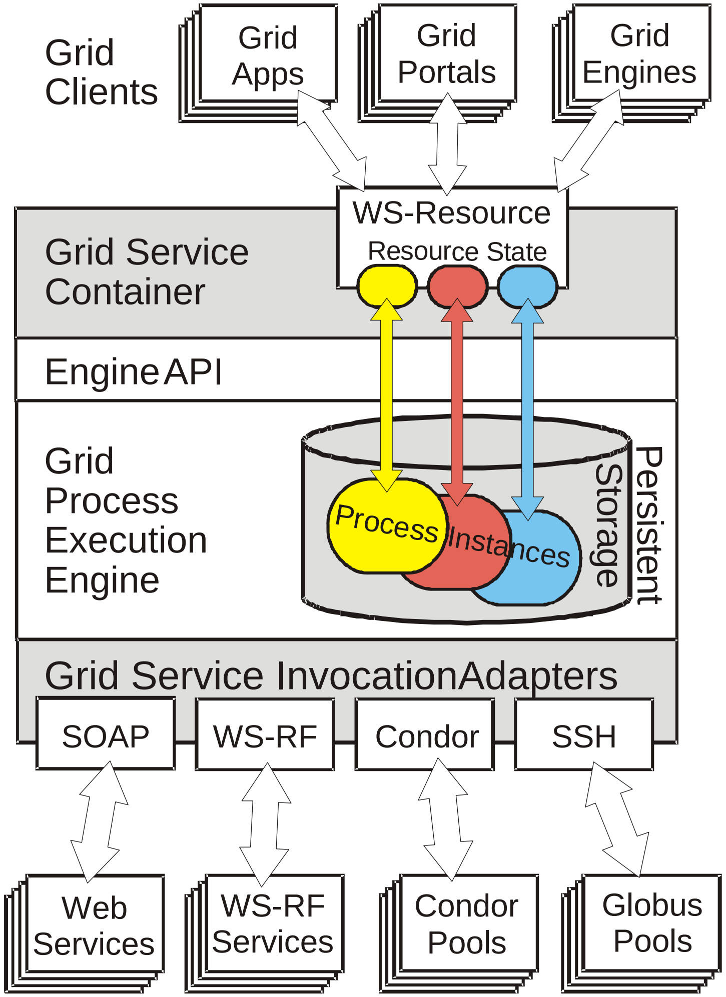
Description	Resource	In Fol
Outbox-Parameter 'Convert	demo.o	de
Process-Output-Parameter '	demo.o	de

- JOpera is a workflow management system providing design, runtime (execution & monitoring) tools for large scale processes

Design

- Design tools let the developer design workflows out of heterogeneous components like Web services, Grid services, Condor, SSH, UNIX programs and many more
- Execution tools allow execution of the workflows in a distributed, autonomic (self-configuring) environment
- Monitoring tools allow the tracking of the execution
- We used Pubscribe and Apollo (code basis WS-Core) as implementation of WS-RF and WS-N

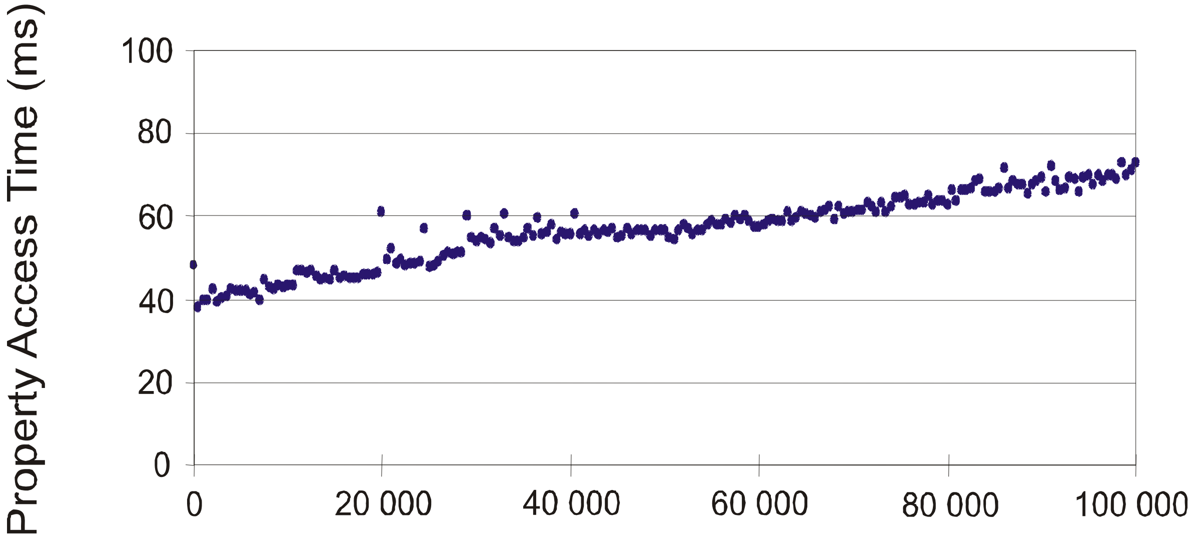
Architecture



Evaluation

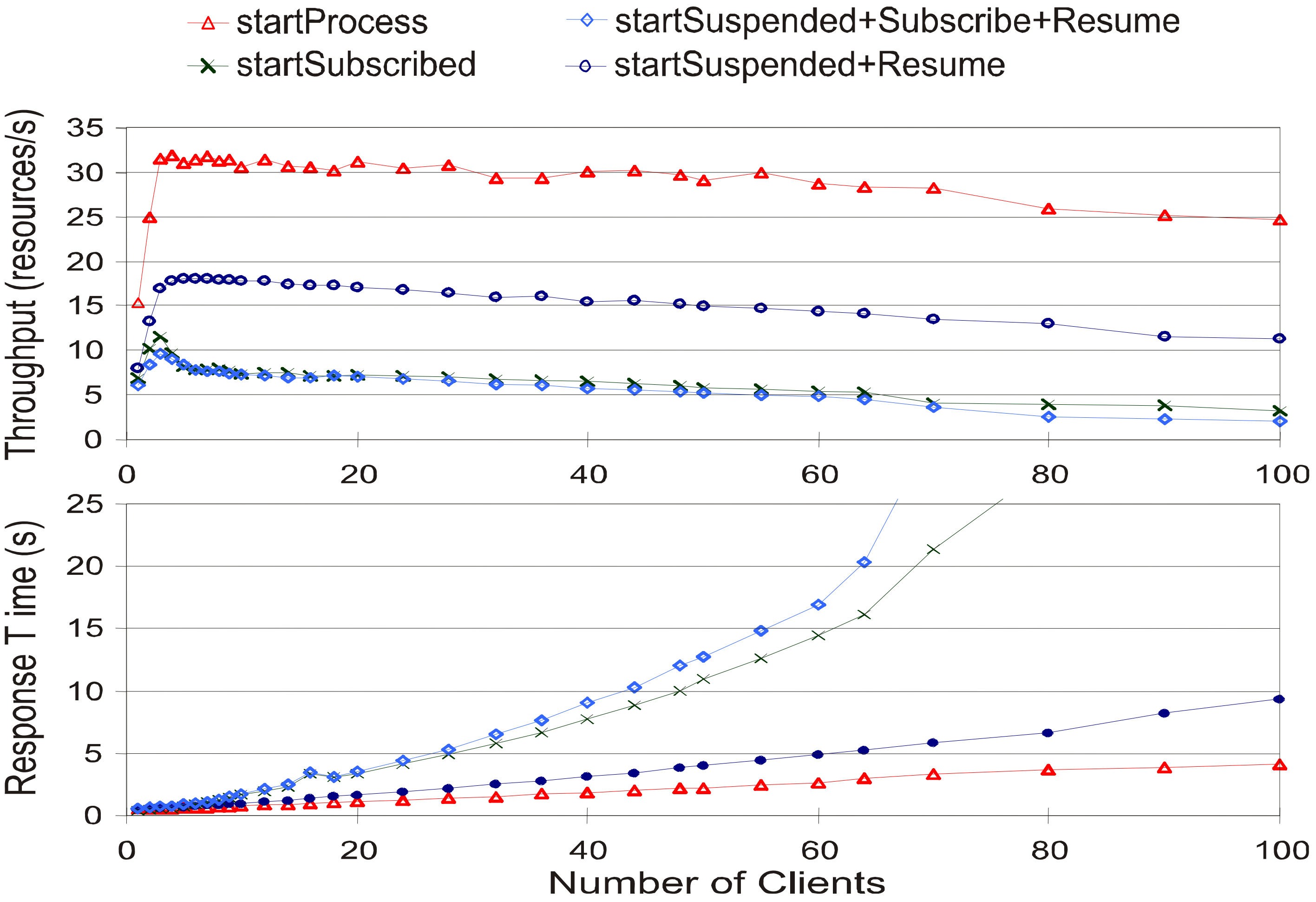
- Querying properties
  - Response time when querying a property for an increasing number of resources
- Resource creation & subscription
  - Response time and throughput for different methods of creating a resource
- Starting process batches
  - Resource creation time for creating each process individually as opposed to start these as a batch

Querying Properties

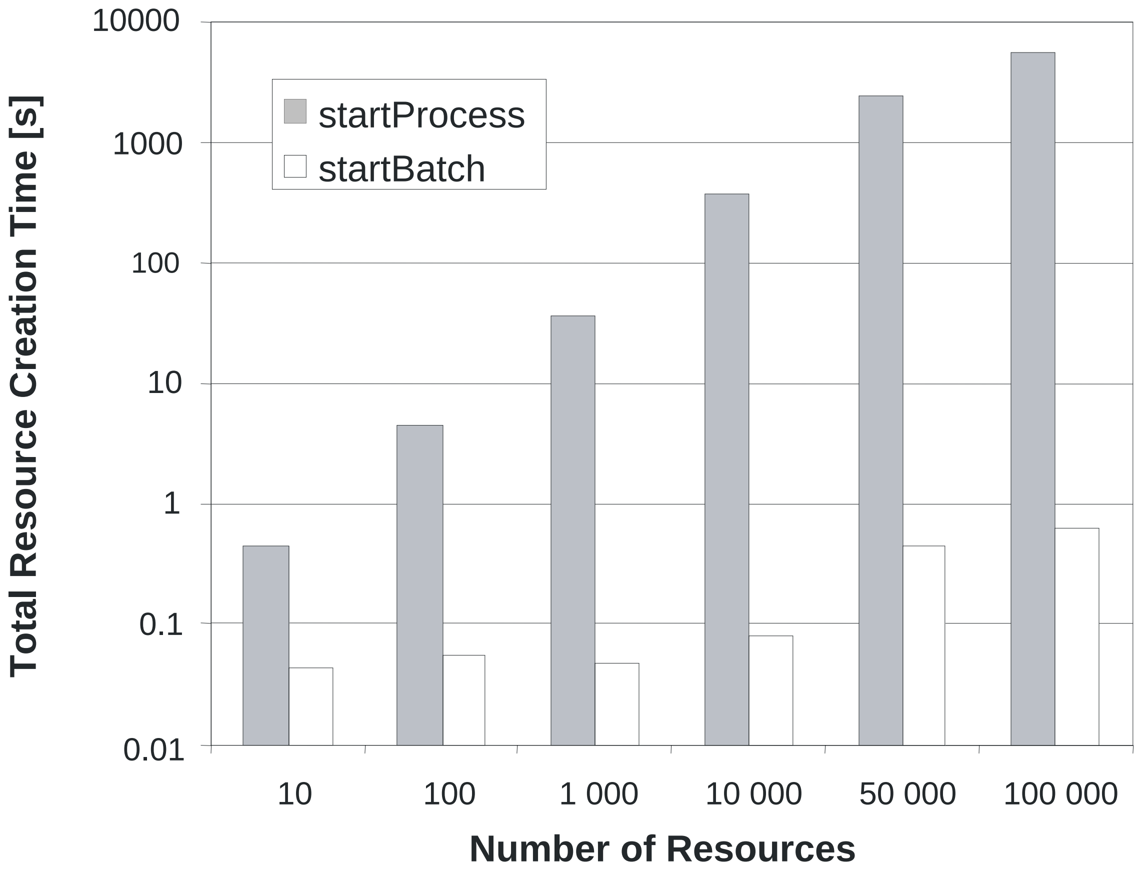


Number of Resources
Number of Process Instances

Resource Creation & Subscription



Starting Process Batches



Conclusions

- With our mapping, we have bridged the gap between processes and resources
- Processes are published as Grid service (WS-RF and WS-N compliant)
- This allows the reuse of Grid service compositions
- Our evaluation shows that the overhead of the mapping layer is minimal and that the system scales to a large number of resources

Publishing Persistent Grid Computations as WS Resources

Thomas Heinis, Cesare Pautasso, Oliver Deak,
Gustavo Alonso

{heinist, pautasso, alonso}@inf.ethz.ch
odeak@student.ethz.ch

ETH Zurich

Jopera Process Support for Web Services available at www.jopera.org