Programming and deployment of scientific workflows with HyperFlow

Bartosz Baliś, Marian Bubak, Kamil Figiela, Maciej Malawski, Maciej Pawlik

AGH University of Science and Technology, Department of Computer Science, Krakow, Poland
AGH University of Science and Technology, Academic Computer Centre Cyfronet AGH, Krakow, Poland
emails: {balis,bubak,kfigiela,malawski}@agh.edu.pl; m.pawlik@cyfronet.pl

HyperFlow: key features
- Simple high-level workflow description + low-level programming capabilities for advanced developers
- Skilled programmers can be as productive as in any mainstream programming language
- Lightweight, non-invasive workflow deployment model that can be applied to various cloud platforms / infrastructures

Conclusions and future work
- HyperFlow can be used both by domain scientists and skilled scientific programmers / software engineers
- Already deployed in various settings and used for a number of scientific applications [1,2]
- Future plans include: scheduler integration, support for more computing infrastructures, research on workflow resilience

Motivation & objectives
- Scientific workflows are arguably the most widely adopted programming model for scientific applications
- Most scientific workflow systems are oriented on domain scientists, offering limited (visual) programming capabilities
- In many cases workflow developers are skilled programmers – they need a productive workflow programming ecosystem

Preparation of a new workflow: main steps
1. Prepare VM images
2. Implement workflow
3. Configure workflow
4. Run!

Rapid deployment model and its benefits
- Workflow management system (WfMS) is treated as part of the cloud application
- Improved isolation: each workflow runs in its own sandbox (separate instance of the workflow runtime system)
- Improved performance: WfMS orchestrates the workflow from inside the cloud leveraging instance-to-instance communication
- Easier integration: no tight coupling integration with a specific cloud platform is required; the same deployment model can be applied to various clouds through deployment plugins
- Cheaper maintenance: new versions of the WfMS can be deployed much faster, no need for thorough software audits

HyperFlow Service deployment architecture (PLGrid example)

Conclusions and future work

Acknowledgements:
This research is supported partly by the European Regional Development Fund program no. POIG.02.03.00-12-137/13 as part of the PLGrid Core; and by AGH grant no. 11.11.230.124.

HyperFlow: workflow graph description example

Processes = workflow activities
- Connected through signals (ins and outs)
- Can be mapped to commands OR JavaScript functions

Workflow graph description example

Conclusions and future work

Conclusions and future work