

Using Scientific Workflows for Science and Engineering Optimisation

David Abramson
University of Queensland
St Lucia, 4072
Australia
+61-7-33656131
david.abramson@uq.edu.au

Keywords

Scientific Workflows; Science Gateways; High Performance Computing; Numerical Optimization.

1. INTRODUCTION

The work described in this extended abstract concerns the synthesis of three normally disconnected pieces of computing infrastructure, namely Scientific Workflows, Engineering Optimization and Science Gateways. When combined, they provide a rich framework for performing engineering design.

Scientific workflows have been applied to a wide range of problems from science and engineering to ecology. They deliver infrastructure that simplifies scripting complex distributed experiments. For example, data may be sourced from one or more locations, and used to drive a pipeline of computational models. Processing steps may vary from simple-minded data reformatting and pre-processing, which can be performed on local workstations, through to computationally intensive models that require supercomputers. Many workflow engines have been produced over the years, and a reasonable summary of these can be found in [8].

Engineering optimization increasingly uses complex computational models that represent some aspects of a system of interest. For example, it can be applied to the problem of finding optimal airfoil shapes as part of an aircraft design. For example, it can be used to compute optimal air pollution control strategies, find optimal shapes for radio antennas, and a wide range of problems. Importantly, optimization algorithms are usually iterative, and when combined with computational models, require repeated executions of a model to produce an “objective value”. This objective value, is then returned to the search algorithm so it can iterate and produce better solutions.

Science Gateways are Web portals that simplify access to complex software services, and may be underpinned by large databases and high performance computers. One of the earliest Science Gateways was NanoHub [6], which provided access to a wide range of engineering design tools, through a simple Web based user interface. Since then, numerous Science Gateways have been built. Traditionally, however, Science Gateways have not supported Scientific Workflows per se, although some do execute workflows behind the gateway as a way of performing computation.

In this keynote address I describe a system that integrates these three technologies, and show how this supports automatic engineering design optimization. Specifically, in the seminar I

will show how it can be applied to airfoil design of very high dimensioned problems.

2. BACKGROUND TECHNOLOGIES

2.1 Kepler

In general, scientific workflows can be data-intensive, compute-intensive, analysis-intensive or visualisation intensive [7]. While there are numerous workflow systems, in this work we have focussed on the Kepler system [9][7][9][5]. Kepler supports different levels of workflows from low-level workflows for grid engineers, to higher-level knowledge discovery workflows for less-technical users. It provides domain scientists with an easy-to-use, yet powerful, system for capturing the workflows they engage with on a daily basis. It streamlines the workflow construction and execution process so that scientists can focus on analyses with minimal effort. Kepler’s actor-oriented modelling is inherited from the Ptolemy II system. Ptolemy II provides module-oriented programming with an emphasis on multiple component interaction semantics. The key principle is to use well-defined Models-of-Computation that govern interactions between components, or actors.

Actors operate like functions in traditional programming languages. Unlike Ptolemy II, Kepler focuses on the design and execution of scientific workflows. Therefore the composition of independent actors forms the scientific workflow.

Kepler’s use of Models-of-Computation, as implemented through “Directors” makes it relatively easy to change the execution semantics. We adopted Kepler because we wanted a more sophisticated execution mechanism, as discussed in Section 2.3. While it would have been possible to add these semantics to other open-source workflow tools, this was a relatively natural extension for Kepler.

2.2 Nimrod

Nimrod enables users to conduct parametric experiments to study behaviours of complex systems [1][2][3][4]. Nimrod supports repeated execution of the same experiments with different input parameters, and it automates several repeated procedures such as formulation, execution, monitoring and result gathering from multiple experiments. Nimrod greatly reduces the programming effort required for experiments, and has a distributed scheduling component. Nimrod focuses on making it easy to repeat such experiments. There are many versions of Nimrod. Here we mention Nimrod/G and Nimrod/O.

Nimrod/G allows users to explore many different scenarios by selecting those that optimise the end results, but it generates an exhaustive search. Nimrod/G can distribute computations to local

computers, remote machines connected by Grid middleware and Cloud resources. The biggest drawback of Nimrod/G when applied to real world engineering problems is that an exhaustive search might be infeasible. Nimrod/O's main goal is to combine rapid application development, distributed computing and optimization into a single tool. Unlike Nimrod/G, however, it uses non-linear optimization techniques to search the outputs of arbitrary computational models. This means that Nimrod/O usually explores many fewer design alternatives than Nimrod/G, making it more efficient. Nimrod/O is, however, able to use Nimrod/G to perform a computation on a remote resource or supercomputer.

2.3 Nimrod/K

As discussed, Nimrod and Kepler both address different aspects of computational science. Kepler makes it easy to specify a single experiment, and Nimrod makes it easy to execute that experiment across different input conditions. We have combined these into Nimrod/K (Nimrod + Kepler). Nimrod/K provides similar functionality to Nimrod/G, but is built on, and extends, Kepler's runtime engine. Thus, it is possible to create arbitrarily complex pipelines, or workflows, of computations, but stream different parameter values through the workflow. By combining Kepler with Nimrod/G, it is possible to run computations on a variety of distributed infrastructure. Likewise, leveraging Nimrod/O's optimization approach makes it possible to search for optimal outputs from a workflow, rather than a single stand-alone computation. Nimrod/K builds on Kepler's standard Directors (SDF and PN), adding a new one for the Tagged Dataflow Architecture (TDA) [1]. The TDA Director builds dynamic concurrency into the workflow and allows independent loops to iterate in parallel.

2.4 Nimrod/OK

Optimization algorithms may themselves be viewed as workflows, usually involving repetitive looping so that results are passed from one iteration to the next. When the features of Nimrod/K and Nimrod/O are combined, optimisation operations are possible – this tool variant is called Nimrod/OK. Nimrod/OK exposes the tasks of an optimization loop and allows the user to assemble novel arrangements of those components. Optimisation algorithms are added as new actors in Kepler, and thus the functionality previously available in Nimrod/O are integrated into Nimrod/OK by building new actors.

2.5 Science Gateways and WorkWays

Science Gateways are Web based portals that hide the complexity of the underlying software and hardware infrastructure. Traditionally, workflows are behind Gateways and are executed as if they are monolithic programs, and results may be rendered in the gateway on completion. This makes it difficult to interact with a pipeline based computation.

WorkWays differs from this by implementing actors that can interact with portal components whilst the workflow is still running. This allows us to gather user input and present output during execution, and even steer the computation as it proceeds. We have demonstrated WorkWays on a number of interactive workflow based computations [10].

3. CONCLUSION

In this keynote I provide more information on the background technologies discussed in Section 2, and show how combining them provides an extremely powerful platform. This platform has the following features:

- Users can express complex computational pipelines using Kepler as a Scientific Workflow Engine. Since Kepler has a large library of pre-existing components, this makes it relatively easy to build complex experiments. Further, Kepler's graphical user interface makes it fairly easy to treat workflows as documentation;
- Nimrod/G can be used to perform computations on remote high end parallel machines. This means that simple actors can be executed locally, but more complex computations, such as engineering models, can be run on supercomputers;
- Nimrod/OK provides the ability to script optimization loops as workflows. Nimrod/OK has a variety of different optimization algorithms that can be matched to the problem at hand;
- WorkWays exposes these workflows through Web technology, allowing a user to both input data to a running optimization workflow and receive information (in graphical form) as to how the computation has proceeded. They can then steer the optimization further.

Below in Figure 1 is a screen capture that illustrates how these combine. In the right hand pane is a Nimrod/K workflow that simulates an aerofoil. The top left image shows a particular design in a 2-dimensioned cross-section. The bottom image shows a Parallel Coordinates visualisation of multiple input parameters and multiple objective function values. These panes are all rendered in the WorkWays web portal, which also allows users to specify and configure the computing resources required to perform the experiment.

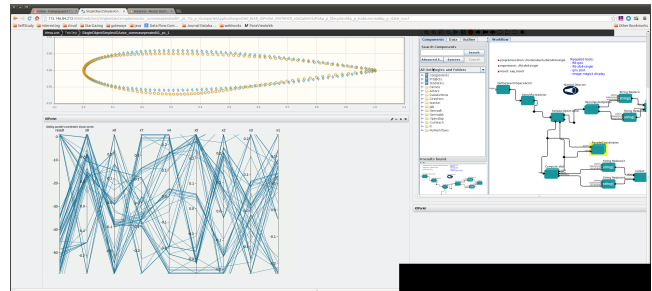


Figure 1 – Optimizing a high dimensional problem in WorkWays

4. ACKNOWLEDGMENTS

Many people and funding bodies have contributed to Nimrod over a significant period. Thanks go to Blair Bethwaite, Colin Enticott, Minh Dinh, Slavisa Garic, Jon Giddy, Chao Jin, Hoang Nguyen and Tom Peachey all of whom contributed to Nimrod/G, Nimrod/K and Nimrod/O. Hoang Nguyen, is responsible for the most recent work on Science Gateways and WorkWays. Timos Kipuros is responsible for recent work on Nimrod/OK and engineering optimization applications.

Funding has been provided by the Australian Research Council and the Distributed Systems Technology Co-operative Research Centre.

5. REFERENCES

- [1] Abramson, D., Bethwaite, B., Enticott, C., Garic, S., Peachey, T., Michailova, A. & Amirrazi, S. 2010. Embedding optimization in computational science workflows. *Journal of Computational Science*, 1, 41-47.

- [2] Abramson, D., Enticott, C. & Altintas, I. 2008. Nimrod/K: towards massively parallel dynamic grid workflows. Proceedings of the 2008 ACM/IEEE conference on Supercomputing. Austin, Texas: IEEE Press.
- [3] Abramson, D., Giddy, J. & Kotler, L. High performance parametric modeling with Nimrod/G: killer application for the global grid? Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International, 2000 2000a. 520-528.
- [4] Abramson, D., Lewis, A. & Peachey, T. 2000b. Nimrod/O: A Tool for Automatic Design Optimization. The 4Th International
- [5] Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B. & Mock, S. Kepler: an extensible system for design and execution of scientific workflows. Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on Scientific and Statistical Database Management, 21-23 June 2004 2004a. 423-424.
- [6] <http://nanohub.org>
- [7] Kepler. 2011. The Kepler Project [Online]. Available: <http://kepler-project.org/> [Accessed 4/11/2011].
- [8] Liu, J., Pacitti, E., Valduriez, P. and Mattoso, M.. A Survey of Data-Intensive Scientific Workflow Management. *J. Grid Comput.* 13, 4 (December 2015), 457-493.
- [9] Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J. & Zhao, Y. 2006. Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18, 1039-1065.
- [10] Nguyen, H., Abramson, D, Kipouros, T, Janke, A and Galloway, G. "WorkWays: Interacting with Scientific Workflows", *Concurrency Computat.: Pract. Exper.*, 27: 4377-4397, 21 May 2015