

Architecting for High Productivity Computing

Simon Cox and Marc Holmes

Summary

Performing a complex computational science and engineering calculation today is more than about just buying a big supercomputer. Although HPC traditionally stands for “High Performance Computing”, we believe that the real end-to-end solution should be about “High Productivity Computing”, by which we mean the whole computational and data handling infrastructure and also the tools, technologies and platforms required by a user to coordinate, execute, and monitor such a calculation end-to-end.

Many challenges are associated with delivering a general high productivity computing (HPC) solution for engineering and scientific domain problems. In this article, we discuss these challenges based on the typical requirements of such problems, propose various solutions, and demonstrate how they have been deployed to users in a specific end-to-end environmental science exemplar. Our general technical solution will potentially translate to any solution requiring controlling and interface layers for a distributed service-oriented HPC service. This article describes our architecture, some of its features and the value it provides.

This architecture has been adopted by dezineforce for its on-demand engineering design service.

Requirements of High Productivity Computing Solutions

In the domains of engineering and science, HPC solutions can be used to crunch complex problems in a variety of areas, such as statistical calculations for genetic epidemiology, fluid dynamics calculations for the aerospace industry, and global environmental modelling. Increasingly, the challenge is in integrating all of the components required to compose, execute, and analyze the results from large-scale computational and data handling problems.

Even with such diverse differences in the problems, the requirements for the solutions have similar features, due to the domain context and the complexity of the problem at hand.

Designed for a solution to a specific problem

Because the calculations and industry involvement are diverse, there are no particular solution providers for any given problem, resulting in highly individualized solutions emerging in any given research department or corporation requiring these calculations. This individuality is compounded by the small number of teams actually seeking to solve such problems and perhaps the need to maintain the intellectual property of algorithms or other aspects of specific processes. Individuality is not in itself an issue—it may be a very good thing—but given that the technical solutions are a

means to an end, it is likely that these individual solutions are not “productized” and thus are probably difficult to interact with or obscure in other ways.

Long-running calculations and processes

The commoditization of the infrastructure required to perform large scale computations and handle massive amounts of data has provided opportunities to perform calculations that were previously computationally impractical. Development of new algorithms and parallelization of code to run across computing clusters can have a dramatic effect, reducing times for computation by several orders of magnitude. So a calculation that would have completed “sometime shortly after the heat death of the universe” could perhaps now run in several weeks or months- which has enabled significant breakthroughs in a variety of industry sectors.

Requirements for provenance information

In many areas of industry, there is a critical need to ensure a useful trail of information for a variety of reasons. There may be simply a need to rerun algorithms and to ensure that the same result sets are produced as a “peace of mind” exercise, but more likely this will be required as part of proof and scientific backup for the publication of research. There may also be statutory reasons for such provenance information: in the aerospace industry, in the event of an air accident investigation, engineers making specific

engineering choices may be liable for the cause of the accident and face criminal proceedings. Therefore, the need to recreate specific states and result sets as required, and to follow the decision paths to such choices is of extreme importance. The complexity of such tasks is compounded when one considers that the life cycle of an aircraft design could be 50 years.

Significant volumes of data and data movement

Significant calculations requiring significant amounts of processing are very likely to involve significant amounts of data throughout the life cycle of the calculation. The operational data sets within the problem space during calculation may be significant. Strategies need to be developed to handle this data, and its metadata. Given the need for provenance information, these strategies need to be both flexible and robust and integrated into the workflow processes used to coordinate the calculations.

Workflow and interfaces

OK, so we have all the parts we need for our solution (hardware for computation and data handling, software and algorithms); but how do we link them together, execute and monitor their execution?

Once computational times have been reduced as far as practical, enhancing the productivity of a solution is much more about consideration of an overall process, which is typically a workflow around the computational tasks. It is also about the development of a solution set to provide user interfaces and controls to allow engineers to carry out their tasks more quickly and efficiently- see figure 1.

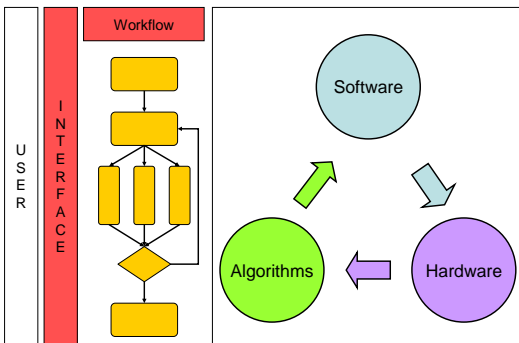


Figure 1 Workflow is the conductor of the user's orchestra; and the interface their front row seat to set up and watch the performance on stage

Case Study: End-to-End High Productivity Environmental Modelling

The GENIE¹ project team is a distributed group of environmental scientists with a common interest in developing and using models to understand the Earth system. GENIE provides a framework that facilitates the integration, execution and management of constituent models for the study of the Earth system over millennial timescales. Earth system simulations are both computationally intensive and data intensive. Simulations based on the GENIE framework need to follow complicated procedures of operations across different models and heterogeneous computational resources.

The simulation codes of varying resolution and complexity for ocean, atmosphere, land surface, sea-ice, ice-sheets, and biogeochemistry are the core simulation *software* for the study along with *algorithms* borrowed from engineering design optimisation to study the response of the system to various climatic forcing events (in the way that an engineer might study the response of a design by changing its dimensions in order to understand or improve its performance.)

The GENIE framework has been designed to support running of such simulations across *hardware* consisting of multiple distributed data and computing resources over a lengthy period of time. GENIE exploits a range of heterogeneous resources, including parallel computing resources running both Linux and Windows Compute Cluster Server and distributed desktop cycle stealing. The GENIE project data store uses Oracle 10G to store metadata about simulations and SQL Server to coordinate the persistence tracking of running workflows- see Figure 2.

At Supercomputing 2006 in Tampa, Fl., it was shown how the use of *workflow* methodology can provide the GENIE simulations with an environment for rapid composition of simulations and a solid hosting environment and to coordinate their execution. The *users* in the collaboration are investigating through the system *interface* how Atlantic thermohaline circulation responds to changes in carbon dioxide levels in the atmosphere and seek to understand, in particular, the stability of key ocean currents under different scenarios of climate change. Their *algorithms* calculate how sea temperature (thermo) and salinity (haline), and density differences drive ocean circulation.

¹ GENIE: Grid-ENabled Integrated Earth System.

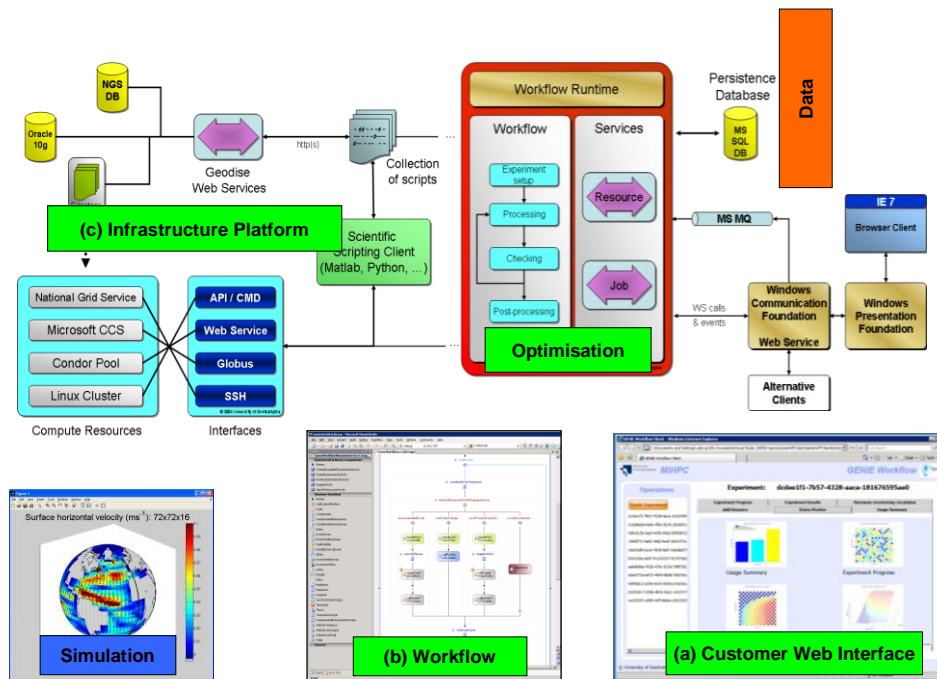


Figure 2 High Productivity Environmental Science end-to-end support: (a) Customer Web Interface using Windows Presentation Foundation (b) Windows Workflow Foundation to author and coordinate simulations, and (c) Heterogeneous Infrastructure consisting of Windows and Linux parallel compute clusters and desktop cycle stealing at distributed sites coupled to Oracle and SQL Server data storage.

[With thanks to Matthew J. Fairman, Andrew R. Price, Gang Xue, Marc Molinari, Denis A. Nicole, Kenji Takeda, and Simon J. Cox (Microsoft Institute of High Performance Computing, School of Engineering Sciences, University of Southampton) External Collaborators: Tim Lenton (School of Environmental Sciences, University of East Anglia) and Robert Marsh (National Oceanography Centre, University of Southampton)]

Component Architecture

If we consider the building blocks of the previous worked example, it looks like a familiar n -tier application architecture and, broadly speaking, this is indeed the case. Figure 3 shows the required components for the architecture.

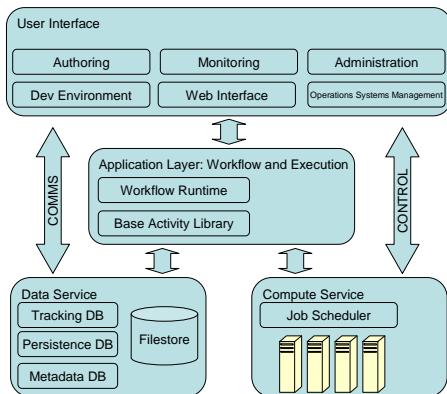


Figure 3 Component Architecture

User Interface

The user interface experience is broken into three parts. Each performs a different function of the overall solution:

- a main user experience for planning, executing and monitoring a computation,
- a workflow authoring experience to configure and link together a set of tasks, and
- an administrative interface for IT administrators to interact with and monitor the health and status of the system. In general this role would not be performed by the system end users.

Application Layer

The application layer consists of the workflow runtime. This layer is used primarily to communicate with the cluster job scheduler but offers a variety of functions:

- an activity and workflow library,

- a rules and policy engine for managing access to a cluster,
- tracking information for job execution providing provenance information as required, and
- persistence information allowing scaling of the solution and providing robustness to long running workflows.

Data Service

The data service contains supporting databases such as tracking and persistence stores for the workflow runtime. In addition there is a metadata store linked to the filestore to enable cataloguing for files in the systems. This database also holds records of available workflows for execution.

Compute Service

The compute service consists of a cluster of compute nodes. This can be as complex as the distributed and

Conclusion

Architecting for high productivity computing is not simply a case of ensuring the “best” performance in order to compute results as quickly as possible—that is more of an expectation than a design feature. The architecture must support the user in all aspects of coordinating, executing and monitoring a complex calculation end-to-end. It must make the system accessible to users; whilst hiding complexity to allow them to focus on their tasks and not the underlying IT infrastructure!

So, how do we recognise “High Productivity Computing” in action? When we are able to *repeatedly* and *effectively* tackle those computational challenges that were previously beyond our reach.

dezineforce has adopted the architecture described in this paper to make computationally intensive design optimisation techniques accessible to engineering companies of all sizes.

Acknowledgements

Peter Williams (Chief Architect, dezineforce)

About the Authors

Simon Cox is Chief Scientist of dezineforce and Professor of Computational Methods in the Computational Engineering Design Research Group within the School of Engineering Sciences of the University of Southampton. He directs the Microsoft Institute for High Performance Computing at the University of Southampton and has published over 100 papers. He can be contacted at simon.cox@dezineforce.com

Marc Holmes is an Architect Evangelist for the Microsoft Technology Centre at Thames Valley Park in the U.K. where he specializes in architecture, design, and proof of concept work with a variety of customers, partners and ISVs. Prior to Microsoft, Marc most recently led a significant development team as Head of Applications and Web at BBC Worldwide. He can be contacted at marc.holmes@microsoft.com.

heterogeneous system in the GENIE case study; or as simple as a homogeneous cluster at a single location. It should be reliable and resilient if high availability is required.

Communication

Communication throughout the architecture is handled using web services to keep the architecture decoupled and scalable, and offers advantages for differing user interfaces based on user requirements (authoring, execution, reporting).

Security

In a distributed service-oriented framework potentially crossing multiple security domains, federation of identities to permit single sign-on and user level access control to all aspects of the component architecture is essential.