

Virtual Microscopy and Analysis using Scientific Workflows

§ David Abramson, § Blair Bethwaite, § Minh Ngoc Dinh, § Colin Enticott, ‡ Stephen Firth, § Slavisa Garic, ‡ Ian Harper, ‡ Martin Lackmann, § Hoang Nguyen, § Tirath Ramdas, § A.B.M. Russel, Ω Stefan Schek, ‡ Mary Vail

§ Faculty of Information Technology,
Monash University,
Clayton, 3800, Victoria, Australia

‡ Faculty of Medicine,
Monash University,
Clayton, 3800, Victoria,
Australia

Ω Leica Microsystems
Am Friedensplatz 3,
Mannheim, D68165, Germany

Abstract - Most commercial microscopes are stand-alone instruments, controlled by dedicated computer systems. These provide limited storage and processing capabilities. Virtual microscopes, on the other hand, link the image capturing hardware and data analysis software into a wide area network of high performance computers, large storage devices and software systems. In this paper we discuss extensions to Grid workflow engines that allow them to execute scientific experiments on virtual microscopes. We demonstrate the utility of such a system in a biomedical case study concerning the imaging of cancer and antibody based therapeutics.

I. INTRODUCTION

Since the advent of the microscope, the development and sophistication of imaging technologies have set the pace in life-sciences, and the ability to “distinguish” smaller and smaller structures continues to define the foundation for biological, bio-medical and bio-technological research. Emerging ultra-high resolution microscopes now allow us to monitor individual molecules in the context of a living cell [23]. Already, off-the-shelf technologies provide real-time, non-invasive images of microscopic bio-medical phenomena, such as the binding of anti-cancer drugs to individual tumour cells, or the trafficking of individually-marked metastasizing tumour cells in and out of blood vessels [13]. However, we expect that both the spatial and temporal resolution of the instruments, and the associated data volumes and rates, will increase dramatically over the next 5 to 10 years. Moreover, our understanding of the underlying cell biology will demand increasingly sophisticated image processing and data mining techniques.

Currently, commercial microscopes are predominantly stand-alone instruments, controlled by dedicated computer systems that can only provide limited storage and processing capabilities. Routinely, the experimenter interacts directly with the microscope, which captures and stores digital images on a local computer disk. The data is then transferred manually to various computer and storage devices for analysis, mining, archiving and visualization. Increasing data volumes are already beginning to create a bottleneck in which the processing times exceed the data acquisition times, significantly impeding real time image visualization and interpretation. Modern image analysis techniques (such as volume rendering), and data mining software, require multi-processor systems for adequate performance and large data stores for file storage, but these require a range of ad-hoc and complicated methods

involving meta-data capture, file transport, job submission and data archiving and visualization.

To address this challenge a number of groups have discussed the seamless integration of image capturing hardware and data analysis software into a wide area network of high performance computers, large storage devices and software systems, generating a ‘virtual’ multi-modal instrument [15][17][5][21]. Such federation supports a variety of different experimental modes, allowing data to be captured on remote instruments, processed using appropriate high performance computers, displayed on a variety of imaging devices and archived on remote storage servers.

Together with our colleagues at Leica, we are building such a Grid environment, depicted in Figure 1. We have implemented a control system that uses a scientific workflow engine to script, and then execute, the experiments using a graphical programming environment. This overall approach has many advantages; most notably the use of a single representation to both document, and execute, an experiment – for the first time, the traditional laboratory notebook becomes executable. Also, because many workflow tools now incorporate the ability to archive, and even share, workflows [16][3], the approach provides a very robust working environment.

However, simply having a workflow engine is not enough to realise the vision. We also need the ability to control and capture images, process them on parallel supercomputers, access network attached storage and display results on visualisation devices.

Workflow tools, such as Kepler, have components that support some of these functions [7]. For example, Grid execution ‘actors’ support remote computation; SRB ‘actors’ provide access to remote Storage Resource Broker services; and local computation actors support simple image manipulation. However, we require additional components to support remote control of microscope hardware, acquisition of images, display of images on a variety of devices, as well as techniques for easily exploiting parallel image processing pipelines. Further, some steps require human guidance and interpretation, so it is impossible to fully automate the image pipeline. To facilitate this we require techniques for involving users in key processing steps within the workflow framework.

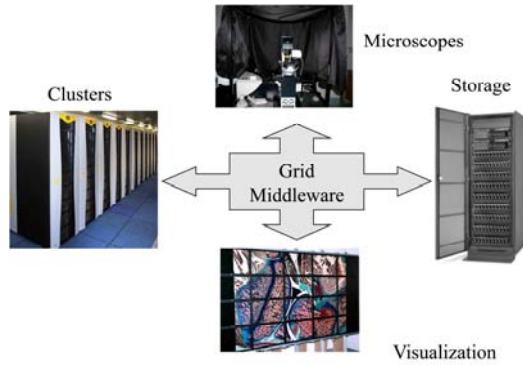


Figure 1 – a Grid enabled virtual microscope

In this paper, we discuss the design and implementation of new Kepler actors that achieve these goals. Specifically, we have built:

- actors for accessing microscopes regardless of their physical location;
- actors that display workflow results on tiled display walls (e.g. OptiPortal [4]);
- actors that incorporate user actions into the workflow.

We also leverage our prior work in providing dynamic parallelism in Kepler, to enable high throughput transparent processing of images using Grid connected parallel machines.

To demonstrate the feasibility of such an approach, we describe an experiment driven by bio-medical science, in which images are captured by microscopes in Australia and Germany, and then processed, displayed and archived in Australia. We leverage the benefits of large tiled display walls by aggregating a number of images, and involve the user in helping evaluate which ones are of interest. The case study highlights the significant advantages in building on open software tools, and illustrates that our approach is flexible and powerful.

II. GRID WORKFLOWS AND KEPLER

Grid workflows allow a user to build arbitrary computations from a set of connected components, or *actors*. Workflows enable applications that control instruments, access and archive data, process the data on high performance computers, display the results on novel visualisation devices and allow the users to interact with their computations. In short, workflows are the scripting language that facilitates the rapid prototyping and execution of novel applications of the type being considered here. Actors may represent computations, but also facilitate access to distributed databases and scientific instruments.

Grid workflow systems allow general applications to be constructed, with examples ranging from ecology to medical imaging. Whilst there are many experimental and commercial workflow systems (see [22] for a fairly recent review paper), in this paper we focus on Kepler [14][7], an open multiparty collaboration. Kepler allows a user to weave a ‘virtual’ application from a set of otherwise distinct components, and its workflow engine orchestrates their execution in a controlled and repeatable manner. Many Kepler features make it an ideal engine for

orchestrating microscopy experiments. For example, Kepler:

- provides a rich set of built-in actors for accessing data and databases;
- provides mechanisms for executing processing logic on local and remote computers;
- provides methods for building web service client-side code, allowing arbitrary web services to be integrated into computations;
- supports a range of different execution semantics, allowing business logic to be separated from performance considerations; and
- is extensible, allowing users to add functionality as required.

On the other hand, it is quite difficult to express workflows with dynamic parallelism using Kepler. This complicates parallel execution of image processing operations, as discussed in the paper. Previously, we have developed a system that augments Kepler with a new *Director* that makes it much easier to specify such experiments [2]. This new system, called Nimrod/K, extends Kepler by providing powerful mechanisms for exposing and managing parallelism in the workflows. In this paper we show how we have used this to process images in parallel, accelerating image throughput.

We have also built a new set of actors that allow workflows to access remote instruments, incorporate user input, and display results on high resolution display devices, such as the OptiPortal. These will be discussed in the next section.

III. WORKFLOW EXTENSIONS

A. New Actors

1) Web service instrument control

Many modern instruments expose their control systems using techniques ranging from direct IP sockets with proprietary protocols through to standard approaches such as RPC, RMI, COM and DCOM. These mechanisms provide a way of controlling the instrument remotely, opening up significant scientific potential as discussed in section 1.

We have used Web services over these mechanisms because they provide an open and extensible interface that can be accessed by a variety of different tools and frameworks. For example, the Kepler workflow engine discussed earlier provides mechanisms for accessing Web services, and thus we can control the microscope from a workflow.

In addition to the backend server side functionality, we have implemented four specific Kepler actors for controlling Leica microscopes. These actors can be used standalone, as part of a larger workflow, or within composite actors. Specifically, they are:

StartScanning: This actor starts the microscope image scanning process using default experimental settings. It checks hardware parameter settings such as focus, objective position, laser, intensity, channel and others before starting image scanning. This process also triggers the conversion of images into different formats, and stores

images at a defined location in the local file system of the microscope (or in a shared file system if one is present). The raw images are generated sequentially channel by channel.

SetDataExporter: This actor sets the default image export URL. This URL is used by the **StartScanning** actor to store images at a defined location. This could be on a network-attached storage based medium, or on a local microscope disk, in which case the file needs to be exposed using some protocol such as FTP, as this then allows the workflow to access the data.

Execute: This actor loads the pre-defined experiment parameters with optimal hardware settings for any experiment. Usually, it is set by an operator, who places a sample, focuses the microscope and stores experiment parameters into an Instrument Parameter Setting (IPS) file. The IPS file consists of all microscope hardware settings that are specific to certain image capture methods.

StopScanning: This actor stops the experiment on the microscope. This actor is important for long running experiments where smart scripting can be used to stop the running experiment and change position of a sample. This mode supports “what if” exploration, and is essential for live cell imaging where cells can move out of focus.

Security is an important consideration in attaching expensive and sensitive instruments to the Grid. There are a number of security models that could be used, however, many of these were unavailable because of security constraints at Leica. For example, WS-Security – the obvious model to be used in conjunction with the Web service interfaces discussed still requires a Web server to be exposed to the Internet. However this requirement clashed with Leica’s security policy so the two were incompatible. Accordingly, we accessed the microscopes via a VPN connection, in which Leica’s corporate network was mounted on the controlling workstation. This allowed us to authenticate using Leica’s approved security mechanism, and the Web server was then exposed using a local IP address. The approach is a little restrictive because it is only possible to access one site in this way. If we needed to VPN to more than one site, the technique would not have worked. An alternative is to use tunnelling techniques, of the type discussed in [20].

2) Displaying results

While microscope images can be rendered using conventional display technology, the output becomes unwieldy when there are either multiple, or single very high resolution, images. For example, image processing operations may generate multiple results (as will be illustrated in our case study), and we may wish to compare and contrast them visually. Alternatively, modern microscopes can scan a large sample, producing tiles that are subsequently stitched into a single large image. Such a requirement demands more screen “real estate” than available on a traditional display.

Recently, DeFanti and others developed the OptiPortal [4], a cluster based system that links many independent monitors to produce a single very large display. The OptiPortal is an ideal display for showing the output of

either a single high resolution image, or multiple independent images. However, to date, there is no easy method for connecting scientific workflows to the OptiPortal, and thus, this is currently a manual and error prone task. Typically, a user must copy the images to the OptiPortal head node, and then use a special windowing environment to display them. This requires the control of multiple systems, and becomes complex when images are produced as a stream.

Accordingly, we have developed a new actor that takes an image object and renders it on the OptiPortal. The actor, **OptiPortalDisplay**, consists of a Java client, modelled on the Kepler **ssh** actor, that connects to the front end of the OptiPortal. It then invokes a special server program, and establishes a direct TCP/IP socket connection. Images are streamed from the client as they are computed, and displayed on the Optiportal. We have based our implementation on the CGLX windowing environment [9], and will expand this to the SAGE platform in the near future [10]. The server currently supports the display of static images and videos, however, we envisage more complex data objects being supported. For example, it should be possible to send a VRML object for display and manipulation using the same actor.

3) Users as actors

Human decision-making capacity is difficult to replace in many applications, including medical imaging. For this reason, it is often necessary to incorporate human feedback within workflows. There are two basic requirements for such functionality: (1) a user interface to allow humans to enter information, and (2) a mechanism to convey this information back into the workflow environment. Furthermore, it is desirable that the user interface is capable of presenting rich information, and also that it is possible to present the interface to a user in a remote location.

Web technology is one implementation approach that can meet all the above basic and desired requirements. Using this approach, a workflow can generate a webpage that presents information in a meaningful manner to aid the decision-making process, and this webpage would convey the user decision through communication with a **WebListener** actor. In the present prototype implementation, the **WebListener** actor binds to a pre-determined port and waits for a HTTP GET request, then conveys user input into the workflow either through the return value (if user input is limited to a small integer range) or through standard output (if user input is less constrained). It is necessary to launch the web interface and **WebListener** in parallel since both processes must run in order for the complete user interface mechanism to work, and this is achieved by running them under Kepler’s Process Networks *Director* (PN).

This mechanism is powerful, and sufficient to support the functionality required for the workflows presented in this paper. Furthermore, the simplicity of this approach allows a large degree of flexibility in the kinds of interactions that may be supported – the tradeoff being that the workflow designer has the burden to ensure that: (1) the webpage interface suits the target application with

the constraint that the user feedback may be expressed as a single HTTP GET request; and that (2) once this information has been conveyed into the workflow, the workflow will process this HTTP GET command and process the information appropriately. Of course, the Web browser used for the interaction can be viewed on the same OptiPortal that shows the images, further enhancing the interaction.

B. Parallel image processing in Kepler

At a functional level, it is possible to specify an arbitrary image processing pipeline using Kepler (and other workflow engines). Operators can either be invoked as local functions (coded in Java) or by remote execution actors using Web services, `ssh` or Grid primitives such as `globus-job-run` (for execution on a single Grid resource) or `nimrod-run` (for meta-scheduling on a set of Grid resources) [1][11].

Running this pipeline in parallel becomes more difficult because Kepler does not provide high-level support for parallel execution of the workflow. Recently we proposed a set of extensions to Kepler, called Nimrod/K, that allow a functional pipeline to run in parallel with minimal changes [2]. Nimrod/K adds a new Kepler *Director* that supports multi-threading. Data tokens are augmented with a tag, or colour, field. When an actor consumes these tokens, a new thread is initiated for each new colour. For example, 100 tokens are sent into an actor, each with a different colour field, then 100 different instances of the actor execute concurrently. If the actor performs a local image processing operation, then 100 processes are forked concurrently on the machine on which Kepler is executing. This would only be efficient if that machine was a symmetric multiprocessor. However, it is also possible to use a remote invocation actor as discussed, in which case, the function can be launched on remote resources, such as Grid connected clusters.

In this paper we will illustrate the combination of the Nimrod/K Director with the `nimrod-run` actor to launch multiple image processing operations on remote clusters, leveraging Nimrod's ability to balance the load across multiple machines.

IV. CASE STUDY

A. Cancer research and imaging

Recently-emerging concepts in tumour therapy are based on the understanding that tumour growth, invasion and metastasis relies on blood vessels for the supply of nutrients, and on connective tissue to provide scaffolding for the tumour cells and blood vessels [12][18][19]. While monoclonal antibodies that target key signalling molecules are already approved anti-cancer agents, their mode of action in tumour vessels is controversial and the subject of intense research.

In this case study we use fluorescent-labelled antibodies together with reagents marking tumour stroma and tumour vasculature to examine the targets of this potential anti-cancer therapeutic. In particular, we use a variety of microscopes and imaging techniques to address a range of research needs.

A "birds-eye" view of the complete, 6 μm -thick tumour section, compiled using the tile scan function of a Leica AF6000LX microscope to image Hoechst-stained cell nuclei and lectin-stained blood vessels, reveals a well-defined tumour centre with tightly-packed tumour cells which is surrounded by extensive vascular network feeding into the tumour. In order to assess if the antibody therapy reaches the vascular and stromal targets within the tumour, we need high-resolution images in selected areas of the tumour section, using a confocal point-scanning microscope. In this case, a Leica TCS SP5 housed at Leica Mannheim in Germany. 2D analysis at four different wavelengths of a selected tumour area within the same section reveals at subcellular resolution the accumulation of the antibody to endothelial cells lining blood vessels and to the connective tissue.

To complete the analysis, we capture on the SP5 a complete optical 'Z-stack' from a 60 μm thick section of the same tumour and use this to recreate a high resolution 3D model of the antibody-targeted tumour vasculature. Before the 3D image can be rendered, it is necessary to 'clean' the data. However, choosing the correct threshold values typically requires user input. The way this is normally performed is for the scientist to apply a threshold, then render the image, and observe if it appears to be correct. If not, then they modify the threshold value and iterate until the image is acceptable. This process can be quite time consuming because a full 3D rendering (taking about 32 minutes) is required at each iteration. As an alternative, we chose to apply a range of different threshold values, render the cleaned images in parallel, and present a range of 3D visualizations (actually as movies) to the operator, who then chooses the most appropriate. The application of high throughput computing techniques reduces the critical path for the operator, at the expense of some additional computation. This approach only works because we have sufficient processors to perform the rendering, and sufficient screen real estate to display the images in one step – making comparison easy.

Overall, the simultaneous display of lower-resolution views of the whole tumour section with high-resolution analyses of areas of interest on a multi-panel screen provides the experimenter with images of unprecedented clarity and resolution. This approach enabled findings about the distribution of the tumour targeting antibody IIIA4 mAb within a tumour – in this particular experiment – with a new level of confidence.

B. The Testbed

As discussed, we make use of two different microscopes, both enabled with the Web Service interface discussed in 3.1.1. Specifically, a Leica AF6000LX wide field microscope at Monash University, Melbourne, Australia and a Leica TCS SP5 confocal microscope at Leica, Mannheim, Germany are used to capture images.

The parallel image processing is performed on a distributed Grid of Linux clusters at various sites around Melbourne. Images are displayed on a 4 x 4 (32 MPixel) OptiPortal at Monash University.

C. The experiment

Here we show a number of Kepler workflows, and describe how these implement the experiment discussed in 4.1. We highlight the use of the new actors and parallel processing capabilities discussed in section 3. In many cases, in order to simplify the discussion in the paper, we have specified high-level *compound* actors that abstract irrelevant detail.

Workflow 1 (Figure 2) shows the logic required to capture the tile-scan from the AF6000LX microscope in Melbourne. The `StartScan` composite actor generates 30 images (two channels by 15 different positions), which are merged to form a two-colour 3x5 tile-scan, covering an area of 2 mm x 4.5 mm of the sample. The workflow copies two directories of files from the microscope (each one containing all positions for that channel), merges them into 15 two colour tiles, stitches the tiles together (using `Tile Placement`) in a pre-defined manner and produces a final bitmap image. It then displays this result on the `OptiPortal` (as shown in Figure 3).

Within this bird's-eye view, one can identify regions of interest for further study (shown by the white circle in Figure 3). A magnified section from within the circle is shown in Figure 4, where the green pixels show a cross section of a blood vessel.

The image in Figure 4 clearly warrants a higher resolution scan rather than a simple magnification of the original tile scan. This functionality is incorporated into Workflow 2 (Figure 5), which displays the image in a web browser with Javascript code to identify the coordinates of a user click within the image. It then transmits these coordinates as a HTTP GET request through a HTTP submission form. This HTTP GET is received by the `WebListener` (discussed in 3.1.2), providing the x and y coordinates selected by the biologist. The coordinates are used to trigger a high-resolution multi-channel scan at the required location. This time, the scan is performed on the SP5 microscope in Germany (using a shipped slide), and the data is retrieved by FTP as before. Importantly, the only change required to the workflow to reflect the different location of the microscope concerns information in the IPS file, discussed in 3.1.1 and the file names for the FTP. The workflow downloads four separate files (one for each channel), and merges these into two separate colour images – one with the Blue and Green channels, and one with all colours. The resulting images are displayed on the `OptiPortal` for examination.

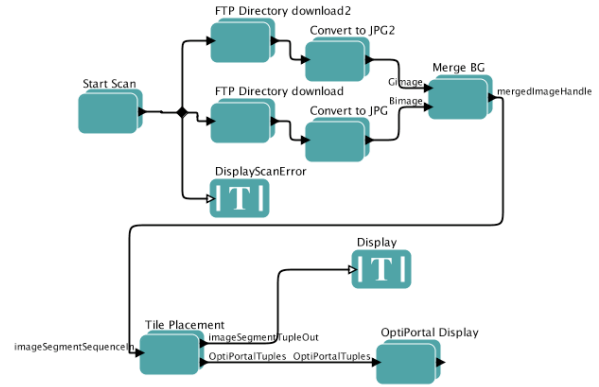


Figure 2 – Workflow 1 – Bird's-eye capture and display

Based on the analysis of the higher resolution scan, shown in Figure 6 (blood vessels, green (RCA-lectin-FITC); tumour targeting antibody, red (labelled with quantum dots)), we then perform a three dimensional z-stack acquisition at the region of interest, again using the SP5 confocal microscope. Workflow 3 (Figure 7) contains the logic discussed in 4.1. Here, 64 different thresholds are applied to the stack to eliminate noise. This is performed as a parameter sweep across the threshold value using ImageJ, an open image processing toolkit commonly used in biomedical research [8]. Since this operation is so fast, we run these on the local workstation. Each of the filtered images is then rendered from 360 single degree perspective views using VolumeJ [6]. This latter operation is performed as a parameter sweep generating 360 different images in total. The resulting 360 * 64 (23,040) images are combined into a movie that shows the results of 64 different threshold values, rotated through 360 degrees, as the movie proceeds. These are presented to a human operator to select the best candidate, and correspondingly the best threshold value.

In order to produce the movie quickly, we leverage the clusters shown in Table 1. The `Nimrod/K Director` and the `NimrodRun` actor are used to parallelise this fairly simple pipeline. Because of the short execution time for individual rendering jobs, we batched them into 20 degree fields, generating a total of 1152 jobs. These were distributed across 517 processors over three Grid resources (East, South and West as shown in Table 1). Using the resources we were able to reduce the rendering time from nearly 34 hours to about 6.5 minutes. Considering the short execution time of each job, and the overheads in scheduling and running Grid jobs, this is an excellent result and compares favourably with a theoretical minimum parallel execution time of 5.3 minutes. Figure 8 shows the execution profile of the run – peaking at 517 parallel jobs. Some of the overheads are visible as a ramp up and ramp down of jobs at the beginning and end of the run.

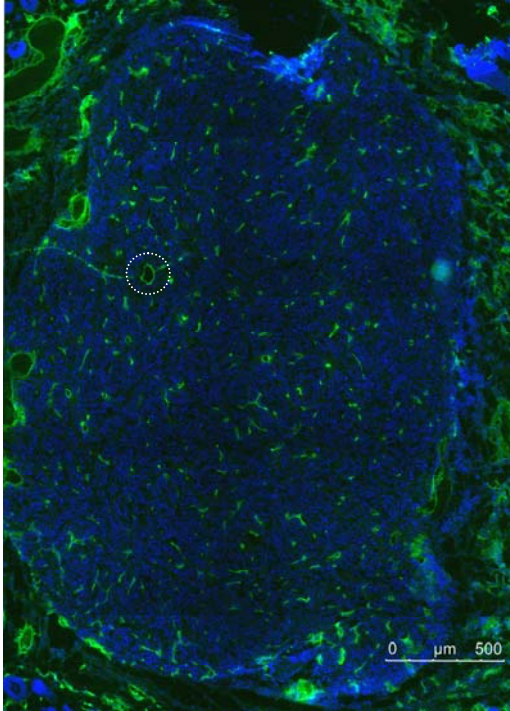


Figure 3 – Bird's-eye view of tumour: cell nuclei in blue (Hoechst stain) and the blood vessels are green (RCA-lectin-FITC). A particular area of interest is marked with a white circle for high- resolution confocal scanning.

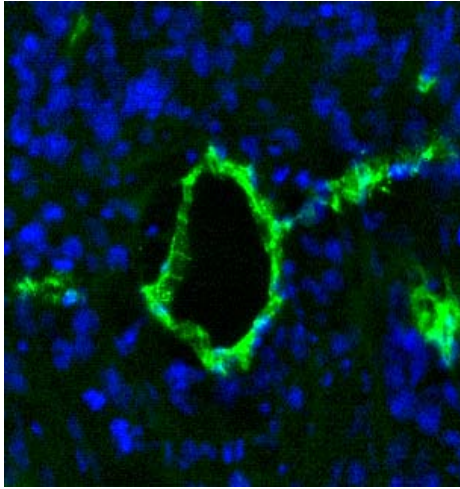


Figure 4 – Magnified area of interest.

The result allows the operator to quickly determine the optimal threshold values, and moreover, see the effect of different values on the resulting image. By examining the movies, it is clear that a number of images contained very little data – this was due to the choice of threshold values for these images. Because all movies are displayed simultaneously, we are able to discard the over-filtered movies very quickly and focus on the ones that produce the best results.

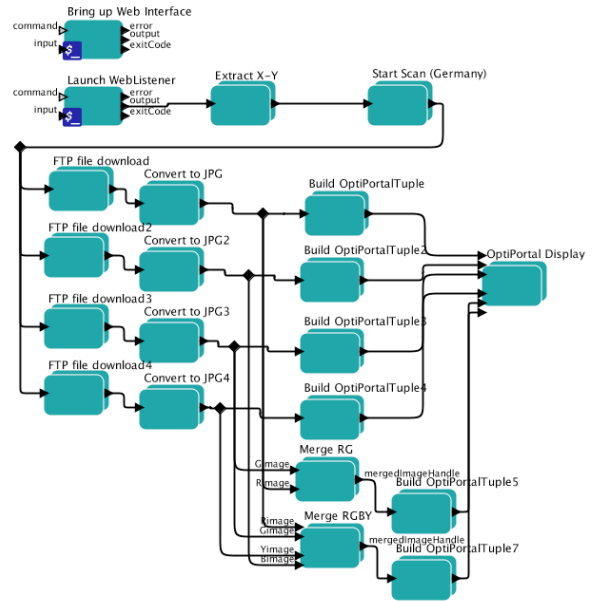


Figure 5 – Workflow 2 – Zooming into area of interest.

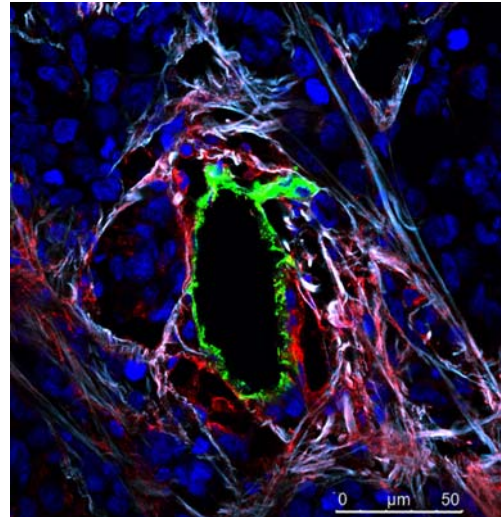


Figure 6 – Output of higher resolution confocal scan. Cell nuclei are blue (Hoechst stain), blood vessels are green (RCA-lectin-FITC), tumour targeting antibody is in red, and connective tissue (anti-tenascin antibody) is in cyan.

V. CONCLUSIONS & FURTHER WORK

The experiment discussed in this paper, and the software we have developed to support it, demonstrate that a new era of experimental science has begun. Users no longer need to be in the same room as their instruments; we have demonstrated that it is possible to control instruments on the other side of the world, yet process, analyse and store the data they produce locally. Moreover, we demonstrate that scientific workflow tools can provide a robust mechanism for specifying and executing experiments. In order to support this, we used our modifications to Kepler that allow it to execute workflow components in parallel on the Grid (Nimrod/K), and we developed new actors that provided missing functionality.

The work to date is only the beginning. While they are powerful, existing workflows contain too many low

level details, making them difficult for non-specialists to use. For example, users must specify operations that copy data across Grid nodes, and this is both complex to create, and difficult to modify. Moreover, complex (but more efficient) transfer patterns, such as third party transfers, are very difficult to specify. Workflows also contain too many domain specific parameters that make them difficult to reuse and share. For example, file system locations, and URLs are typically coded in the workflow. Accordingly, we are developing a data transport layer for Nimrod/K that abstracts the way data is moved around the Grid.

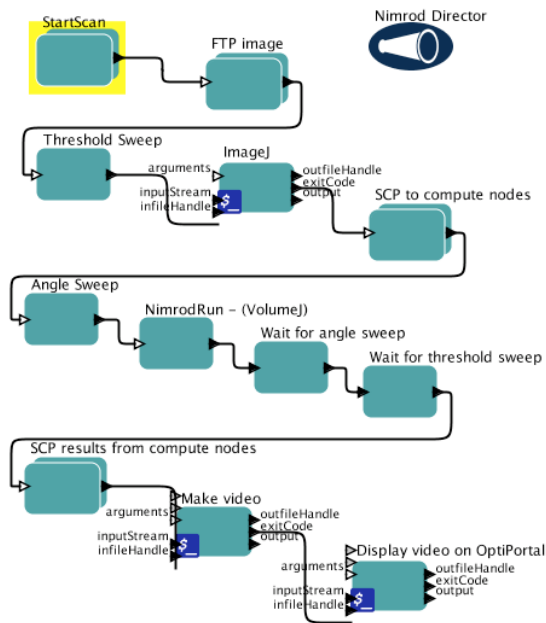


Figure 7 – Workflow 3 – Image cleanup and rendering

Our current implementation requires many enhancements before it can be used routinely. For example, we need to enhance the functionality in libraries like CGLX and SAGE in order to create a more flexible working environment. Whilst the OptiPortal Display actor has demonstrated that it is possible to link a workflow engine with the imaging environment, this link needs to support more data types and support a more interactive mode of operation – currently the libraries are biased towards the separate production of images and their display.

The Web Service APIs we have specified to the Leica microscopes are simple, and require enhancement to enable more control of the instruments. At present, they mimic the manual mode of operation, but we need to expose the significant complexity of the basic instruments by a powerful, but safe, set of APIs. We will further develop these over the coming months.

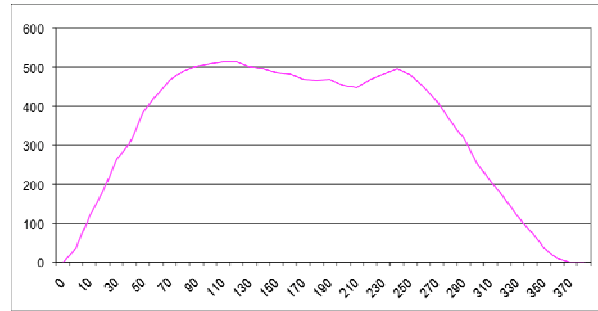


Figure 8 – Execution profile for rendering jobs

ACKNOWLEDGEMENTS

The Nimrod/K project is supported by an Australian Research Council (ARC) Discovery grant. The collaboration between Monash University and Leica Microsystems is supported by an ARC Linkage grant. Hoang Nguyen was supported under Monash Undergraduate Research Projects Abroad (MURPA) funding.

REFERENCES

- [1] Abramson, D. Giddy, J. and Kotler, L. "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid," In *Int'l. Parallel and Distributed Processing Symposium (IPDPS)*, Cancun, Mexico, May 2000.
- [2] Abramson, D., Enticott, C and Altinas, I. "Nimrod/K: Towards Massively Parallel Dynamic Grid Workflows", IEEE Supercomputing 2008, Austin, Texas, November 2008.
- [3] De Roure, D., Goble, C. and Stevens, R. (2009) "The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows". *Future Generation Computer Systems* 25, pp. 561-567. doi:10.1016/j.future.2008.06.010.
- [4] DeFanti, T. et al. (2009) The OptiPortal, a scalable visualization, storage, and computing interface device for the OptiPuter", *Future Generation Computer Systems*, Volume 25, Issue 2, February 2009, Pages 114-123, ISSN 0167-739X, DOI: 10.1016/j.future.2008.06.016.
- [5] Görtler, J. Berghoff, M. Kayser, G. and Kayser, K. "Grid technology in tissue-based diagnosis: fundamentals and potential developments". *Diagnostic Pathology*, 2006.
- [6] <http://bij.isi.uu.nl/vr.htm>
- [7] <http://kepler-project.org>
- [8] <http://rsb.info.nih.gov/ij/>
- [9] <http://vis.ucsd.edu/~cglx/main.html>
- [10] <http://www.evl.uic.edu/cavern/sage/index.php>
- [11] <http://www.messagelab.monash.edu.au/nimrod>.
- [12] Jain, R. K. (2005) Normalization of tumor vasculature: an emerging concept in antiangiogenic therapy. *Science* 307: 58-62.
- [13] Jain, R.K., Munn, L.L., Fukumura, D., 2002, *Nature Rev. Cancer* 2, 266-276
- [14] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao J. and Zhao, Y. "Scientific Workflow Management and the Kepler System", *Concurrency and Computation: Practice & Experience*, Special Issue on Scientific Workflows, 2005.
- [15] Lundin M, Lundin J, Isola J. "Virtual Microscopy: applications in diagnostic pathology", *Journal of Clinical Pathology*. 2004;57:1250-1251.
- [16] Oinn, T. Addis, M. Ferris, J. Marvin, D. Senger, M. Greenwood, M. Carver T. and Glover, K. Pocock, M.R. Wipat, A. and Li. P. "Taverna: a tool for the composition

- and enactment of bioinformatics workflows”, *Bioinformatics*, 20(17):3045-3054, Oxford University Press, London, UK, 2004.
- [17] Pan, T. Jewel, S. Catalyurek, U. Wenzel, P. Leone, G. Hastings S., Oster, S. Langella, S. Kurc, T. Saltz, J. and Cowden, D. “Virtual Microscopy: Distributed Image Storage, Retrieval, Analysis, and Visualization”, *Parallel Computing for Bioinformatics and Computational Biology*, 2006.
 - [18] Polyak, K, Weinberg, R.A. (2009) Transitions between epithelial and mesenchymal states: acquisition of malignant and stem cell traits. *Nat Rev Cancer* 9: 265-273.
 - [19] Psaila, B., Lyden, D. (2009) The metastatic niche: adapting the foreign soil. *Nat Rev Cancer* 9: 285-293.
 - [20] Tan, J., Abramson, D. & Enticott, C. “REMUS: A Rerouting and Multiplexing System for Grid Connectivity Across Firewalls”, *Journal of Grid Computing*, Online version doi:10.1007/s10723-008-9104-1, June 2008.
 - [21] Yang, L. Chen, W. Meer, P. Salaru, G. Goodell, L. A. Berstis, V. and Foran D. J., “Virtual Microscopy and Grid-enabled Decision Support for Large Scale Analysis of Imaged Pathology Specimens”. *IEEE Transactions on Information Technology in Biomedicine*, 2009.
 - [22] Yu, J. and Buyya, R. “A Taxonomy of Workflow Management Systems for Grid Computing,” *Technical Report GRIDS-TR-2005-1*, Grid Computing and Distributed Systems Laboratory, Univ of Melbourne, 2005. <http://www.gridbus.org/reports/GridWorkflowTaxonomy.pdf>.
 - [23] Yuste, R., 2005, *Nature Methods* 2, 902-4.