A Survey of Distributed Workflow Characteristics and Resource Requirements

Lavanya Ramakrishnan
Department of Computer Science, School of Informatics
Indiana University, Bloomington, IN

Dennis Gannon
Microsoft Research, Redmond, WA
Department of Computer Science, School of Informatics
Indiana University, Bloomington, IN

Abstract

Workflows have been used to model repeatable tasks or operations in a number of different industries including manufacturing and software. In recent years, workflows have increasingly used distributed resources and web services through resource models such as grid and cloud computing. These workflows often have disparate requirements and constraints that need to be accounted for during workflow orchestration. In this paper, we present workflow examples from different domains including bioinformatics and biomedical, weather and ocean modeling, astronomy detailing their data and computational requirements.

1 Introduction

Workflows and workflow concepts have been used to model a repeatable sequence of tasks or operations in different domains including the scheduling of manufacturing operations, inventory management, etc. The advent of internet and web services has seen the adoption of workflows as a means for business process management [31] and as an integral component of cyberinfrastructure for scientific experiments [11, 17]. In addition, the availability of distributed resources through grid and cloud computing models has enabled users to share data and resources using workflow tools and other user interfaces such as portals.

Workflow tools allow users to compose and manage complex distributed computation and data in distributed resource environments. Workflows might have different resource requirements and constraints associated with them. For example, application workflows with stringent deadline driven requirements such as weather prediction, economic forecasting are now increasingly run in distributed resource environments.

In this paper we discuss workflow examples from different domains including bioinformatics and biomedicine, weather and ocean modeling, astronomy, etc. These examples have been obtained by talking to domain scientists and computer scientists who composed and/or run these workflows. Each of these workflows have been modeled using different workflow tools and sometimes the flow is even managed through scripts. For each workflow we specify the running time of applications and input and output data sizes associated with each task node. Running time of applications and data sizes for a workflow depend on a number of factors including user inputs, specific resource characteristics and run-time resource availability variations [20]. Thus our numbers are approximate estimates for typical input data sets that are representative of the general characteristics of the workflow.

In the following sections, we provide a brief description of the project, workflow and usage model of the workflows as available today. For each of the workflows, we also provide a DAG representation of the workflow annotated with computation and data sizes. In addition the project and organization names and contact person for the workflows are specified. This is not a complete list but represents the contributions by the individuals and organizations that responded to the survey request.

2 Weather and Ocean Modeling

In the last few years the world has seen a number of severe natural disasters such as hurricanes, tornadoes, floods, etc. The models used to study weather and ocean phenomenon use real-time observational data in conjunction with a number of parameters that are varied to study the possible scenarios for prediction. In addition the models must be run in a timely manner and information disseminated to disaster response agencies. This creates the need for *large scale modeling* in the areas of meteorology and ocean sciences, coupled with an *integrated environment* for analysis, prediction and information dissemination. A number of cyberinfrastructure projects are building tools and constructing workflows to facilitate next-generation weather and ocean modeling science.

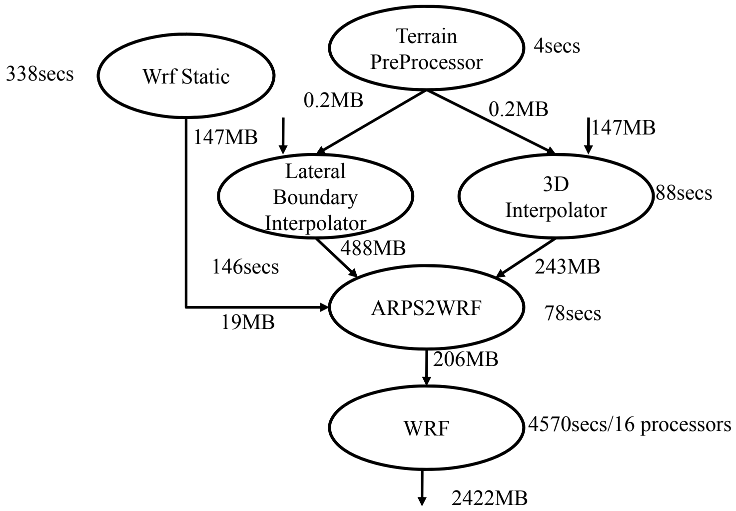

Figure 1: LEAD NAM initialized forecast workflow. The workflow processes terrain and observation data to produce weather forecasts.

2.1 Mesoscale Meteorology

Project: Linked Environments for Atmospheric Discovery, TeraGrid Science Gateway
Websites: `http://portal.lead.project.org`
Tool: xbaya, GPEL, Apache ODE
Description: The Linked Environments for Atmospheric Discovery (LEAD) [18] is a cyberinfrastructure project that supports mesoscale meteorology. The infrastructure of LEAD needs to support real-time dynamic, adaptive response to severe weather. A LEAD service workflow has constraints on execution time and accuracy due to weather prediction deadlines. The typical inputs to a workflow of this type are streaming sensor data @w [18, 27] that must be pre-processed and then used to launch an ensemble of weather models. The model

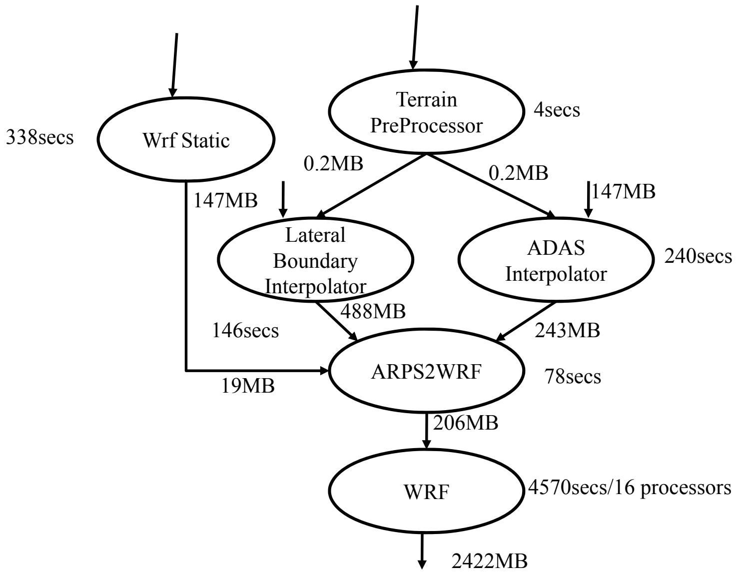

Figure 2: LEAD ADAS initialized forecast workflow. The workflow processes terrain and observation data to produce weather forecasts.

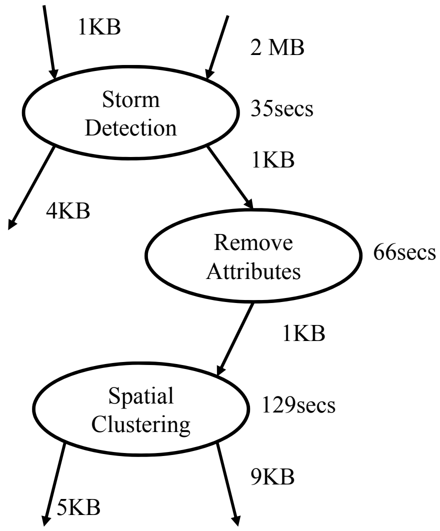

Figure 3: LEAD Data Mining Workflow workflow. The workflow processes weather data to identify regions where weather phenomenon might be present.

outputs are processed by a data mining component that determines whether some ensemble set members must be repeated to realize statistical bounds on prediction uncertainty. Figures 1, 2 and 3 show the workflows available through the LEAD portal that include weather forecasting and data mining workflows [22]. The weather forecasting workflows are largely similar and vary only in their preprocessing or initialization step. While the data mining workflow can be run separately today, it can trigger forecast workflows and/or steer remote radars for additional localized data in regions of interest [27].

Use case scenarios: The LEAD workflows are used in a number of different scenarios - a new workflow might be initiated in response to a weather condition or a number of workflows might be launched as part of an educational workshop. In LEAD, the user might want to *specify constraints* to adjust the forecast configuration based on resource availability [30].

User-initiated Workflows. The typical mode of usage of the LEAD system is where a user logs into the portal and to launch a workflow for some analysis. The user selects a pre-composed workflow and supplies the necessary data for the run. In this scenario, we need mechanisms to procure resources and enable workflow execution, provide recovery mechanisms from persistent and transient service failures and adapt to resource availability or recover from resource failures during workflow execution. The user might also want the ability to pause the workflow at the occurrence of a predefined event, inspect intermediate data and make changes during workflow execution.

Workflow Priorities Let us consider a scenario of an educational LEAD workshop with multiple competing users. Resources are typically reserved for this event through out-of-band mechanisms for advanced reservation. In this scenario resource allocation needs to be based on existing load on the machines, resource availability, the user priorities and workflow load. The bounded set of resources available to the workshop might need to be proportionally shared among the workflow users. If there is a weather event during the workshop, resources might need to be reallocated and conflicting events might need some arbitration.

Dynamic Weather Event. Let us consider a case of dynamic weather event detection with data mining. Multiple weather events and their scale might need priorities between users for appropriate allocation of limited available resources. Resources must be allocated to meet deadlines. Additionally, to ensure successful completion of tasks, we might need to replicate some of the workflow tasks for increased fault tolerance. It is possible that with advance notice of upcoming weather events, we might want to anticipate the need for resources and try to procure them in advance.

Advanced User Workflow Alternatives and Constraints. In LEAD, an advanced user might have a workflow and in addition provide a set of constraints (e.g. time deadline). For example the user might specify that fewer parallel ensemble members could be run to get a quicker result. Alternatively the user might be willing to sacrifice forecast resolution to get some early results which might then define the rest of the workflow.

These scenarios illustrate the need for an adaptation framework that implements *online planning and control of workflows* to assess resource needs, proactively adapt to failures and workflow needs based on priorities and policies specified by the user.

2.2 Storm surge modeling

Project: Southeastern Coastal Ocean Observing and Prediction Program (SCOOP)
Contact: Brian Blanton, Howard Lander, Steve Thorpe
Organization(s): Renaissance Computing Institute
Websites: `http://www.renci.org/focusareas/disaster/scoop.php`
Tool: [Scripts]

Description: Southeastern Universities Research Association's (SURA) Southeastern Coastal Ocean Observing and Prediction (SCOOP) program is a distributed project that is creating an open-access grid environment for the southeastern coastal zone to help integrate regional coastal observing and modeling systems [7, 28].

Storm surge modeling requires assembling input meteorological and other data sets, running models, processing the output and distributing the resulting information. In terms of modes of operation, most meteorological and ocean models can be run in hindcast mode, as an after fact of a major storm or hurricane, for post-analysis or risk assessment, or in forecast mode for prediction to guide evacuation or operational decisions [28]. The forecast mode is driven by real-time data streams while the hindcast mode is initiated by a user. Often it is necessary to run the model with different forcing conditions to analyze forecast accuracy. This results in a large number of parallel model runs, creating an ensemble of forecasts. Figure 4 shows a five member ensemble run of tidal and storm-surge ADCIRC [24] model. For increased accuracy of forecast the number of concurrent model runs might be increased. ADCIRC is a finite element model that is parallelized using Message Passing Interface (MPI).

The SCOOP ADCIRC workflows are launched according to the typical six hour synoptic forecast cycle used by the National Weather Service and the National Centers for Environmental Prediction (NCEP). NCEP computes an atmospheric analysis and forecast four times per day at six hour intervals. Each of the member runs i.e. each branch of the workflow gets triggered when wind files arrive through Local Data Manager (LDM) [8], an event-driven data distribution system that selects, captures, manages and distributes meteorological data products. The outputs from the individual runs are synthesized to generate the workflow output that is then distributed through LDM.

In the system today each arriving ensemble member is handled separately through a set of scripts and Java code [28]. The resource selection approach [21] makes a real-time decision for each model run and uses knowledge of scheduled runs to load-balance across available systems. However this approach does not have any means of guaranteeing desired QoS in terms of completion time.

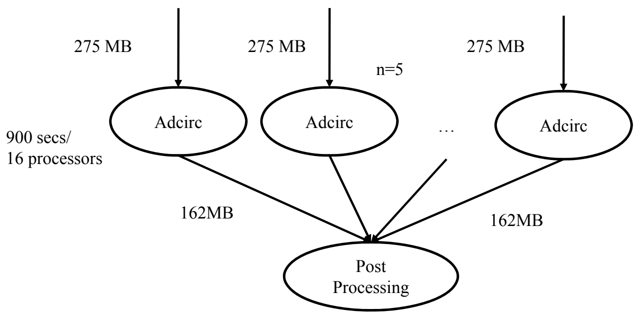

Figure 4: SCOOP workflow. The arriving wind data triggers ADCIRC that is used for storm-surge prediction during hurricane season.

2.3 Floodplain Mapping

Project: North Carolina Floodplain Mapping Program
Contact: Howard Lander, Brian Blanton
Organization(s): Renaissance Computing Institute
Tool: [Scripts]
Description: The North Carolina Floodplain Mapping Program [5, 12] is focused on developing accurate

simulation of storm surges in the coastal areas of North Carolina. The deployed system today consists of a four-model system that consists of the Hurricane Boundary Layer (HBL) model for winds, WaveWatch III and SWAN for ocean and near-shore wind waves, and ADCIRC for storm surge. The models require good coverage of the parameter space describing tropical storm characteristics in a given region for accurate flood plain mapping and analysis. Figure 5 shows the dynamic portion of the workflow. Forcing winds for the model runs are calculated by the Hurricane Boundary Layer(HBL) model that serve as inputs to the workflow. The HBL model is run on a local commodity linux cluster. Computational and storage requirements for these workflows are fairly large requiring careful resource planning. The rest of the workflow today runs on RENCI's Bluegene system [6].

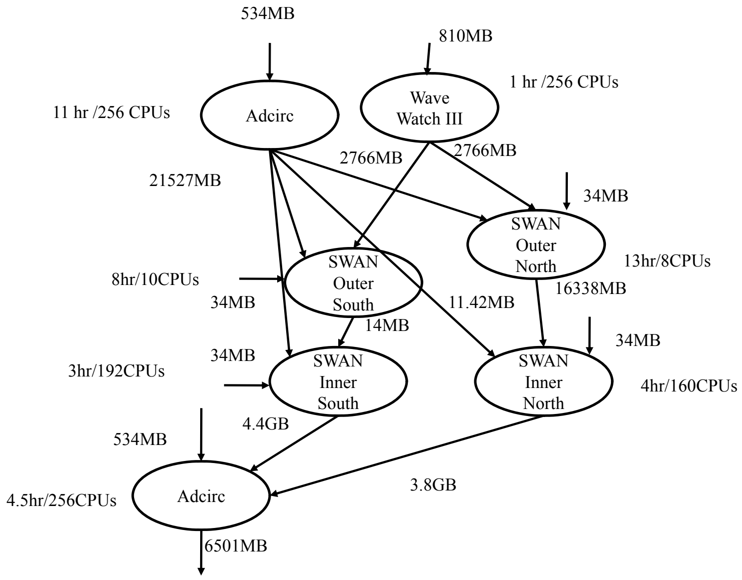

Figure 5: NCFS workflow. A multitude of models are run to model the storm surges in the coastal areas of North Carolina.

3 Bioinformatics and Biomedical workflows

The last few years have seen large scale investments in cyberinfrastructure to facilitate Bioinformatics and biomedical research. The infrastructure allows users to access databases and web services through workflow tools and/or portal environments. We surveyed three major projects in the United States - North Carolina Bioportal, cancer Biomedical Informatics Grid (caBIG), and National Biomedical Computational Resource (NBCR) to understand the needs of this class of workflows. Significant number of these workflows involve small computation but involve access to large-scale databases that need to be preinstalled on available resources. While the typical use cases of today have input data sizes in the order of megabytes, it is anticipated that in the future data sizes might scale to gigabytes.

3.1 Glimmer

Project: North Carolina Bioportal, TeraGrid Bioportal Science Gateway
Organization(s): Renaissance Computing Institute
Websites:

https://portal.renci.org/portal/
http://www.renci.org/focusareas/biosciences/motif.php
http://www.motifnetwork.org/
Tool: Taverna
Description: The North Carolina Bioportal and The TeraGrid Bioportal Science Gateway [29] provides access to about 140 bioinformatics applications and a number of databases. Researches and educators use the applications interactively for correlation, exploratory genetic analysis, etc. The Glimmer workflow is one such example workflow that is used to find genes in microbial DNA (Figure 6).

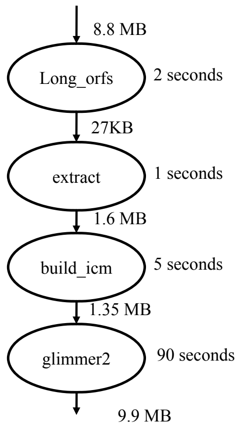

Figure 6: Glimmer workflow. A simple workflow used in educational context to find genes in microbial DNA.

3.2 Gene2Life

Project: North Carolina Bioportal, TeraGrid Bioportal Science Gateway
Organization(s): Renaissance Computing Institute
Websites:
https://portal.renci.org/portal/
http://www.renci.org/focusareas/biosciences/motif.php
http://www.motifnetwork.org/
Tool: Taverna
Description: Let us consider the Gene2Life workflow used for molecular biology analysis. This workflow takes an input DNA sequence, searches databases to find genes matching the sequence. It globally aligns the results and attempts to correlate the results based on organism and function. Figure 7 depicts the steps of the workflow and the corresponding output at each stage. In this workflow the user provides a sequence that can be a nucleotide or an amino acid. The input sequence performs two parallel BLAST [10] searches, against the nucleotide and protein databases respectively. The results of the searches are parsed to determine the number of identified sequences that satisfy the selection criteria. The outputs trigger the launch of ClustalW, a bioinformatics application that is used for the global alignment process to identify relationships. These outputs are then passed through parsimony programs for analysis. The two applications that may be available for such analysis are dnapars and protpars. In the last step of the workflow plots are generated to visualize the relationships, using an application called drawgram.

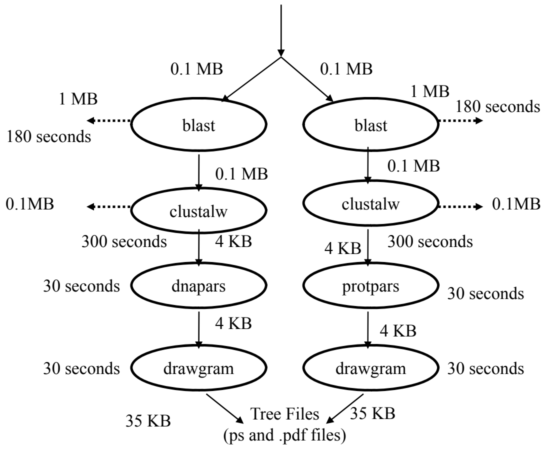

Figure 7: Gene2Life workflow. The workflow is used for molecular biology analysis of input sequences.

3.3 Motif Network

Project: Motif Network
Contact: Jeffrey Tilson
Organization(s): Renaissance Computing Institute
Websites:
`http://www.renci.org/focusareas/biosciences/motif.php`
`http://www.motifnetwork.org/`
Tool: Taverna
Description: The MotifNetwork project [32, 33], a collaboration between RENCI and NCSA, is building a software environment to provide access to domain analysis of genome sized collections of input sequences. The MotifNetwork workflow is computationally intensive. The first stage of the workflow assembles input data and processes the data that is then fed into Interproscan service. The concurrent executions of InterProScan is handled through Taverna and scripts. The results of the domain “scanning” step are passed to an MPI code for the determination of domain architectures.

3.4 MEME-MAST

Project: National Biomedical Computation Resource (NBCR)
Contact: Sriram Krishnan
Organization(s): San Diego Supercomputing Center (SDSC)
Websites: `http://nbcr.sdsc.edu/`
Tool: Kepler
Description: The goal of National Biomedical Computation Resource(NBCR) is to facilitate biomedical research by harnessing advanced computational and information technologies. The MEME-MAST (Figure 9) workflow deployed using Kepler [9, 23] allows users to discover signals or motifs in DNA or protein sequences and then search the sequence databases for the recognized motifs. This is a simple workflow often used for demonstration purposes.

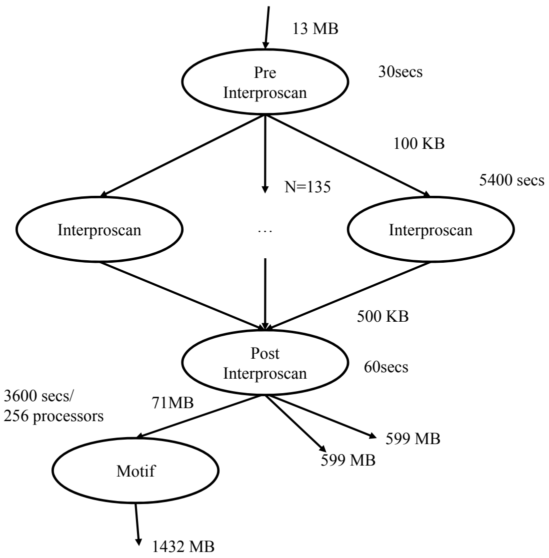

Figure 8: Motif workflow. A workflow used for motif/domain analysis of genome sized collections of input sequences.

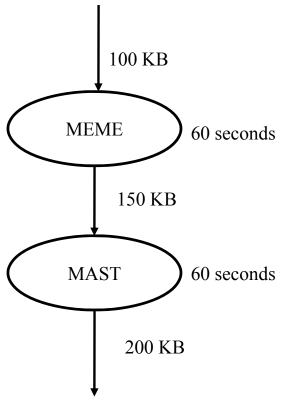

Figure 9: MEME-MAST workflow. A simple demonstration workflow used to discover signals in DNA sequences.

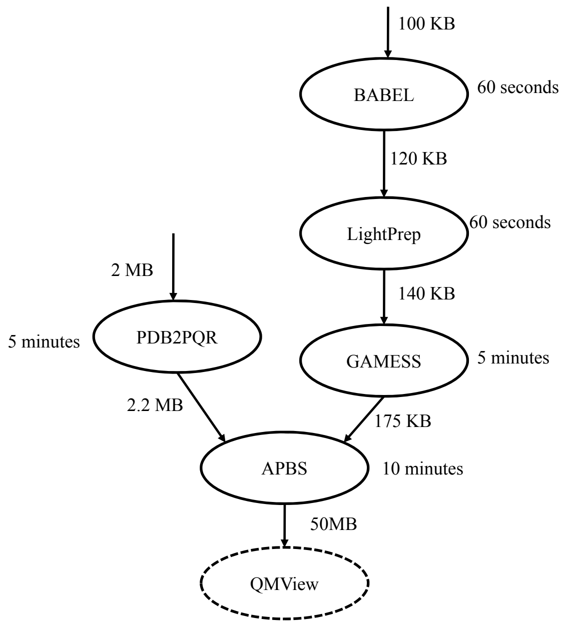

Figure 10: Molecular Sciences workflow. The workflow is used to study atomic structures of proteins and ligands.

3.5 Molecular Sciences

Project: National Biomedical Computation Resource (NBCR)
Contact: Sriram Krishnan
Organization(s): San Diego Supercomputing Center (SDSC)
Websites: `http://nbcr.sdsc.edu/`
`http://gemstone.mozdev.org`
Tool: Gemstone
Description: An important process in the drug-design process is understanding the three-dimensional atomic structures of proteins and ligands. The Gemstone project, a client interface to a set of computational chemistry and biochemistry tools, provides the NBCR community access to a set of tools that allows users to analyze and visualize atomic structures. Figure 10 shows an example molecular science workflow. The workflow in its current incarnation runs in an interactive mode where each step of the workflow is manually launched by the user once the previous workflow task finishes. The first few steps of the workflow involve downloading the desired protein and ligand from the Protein Data Bank (PDB) database and converting it to a desired format. Concurrent preprocessing is done on the ligand using the Babel and LigPrep services. Finally GAMESS and APBS are used to analyze the ligand and protein. The results are finally visualized using the QMView which is done as an offline process.

3.6 Avian Flu

Project: National Biomedical Computation Resource (NBCR), Avian Flu Grid, Pacific Rim Application and Grid Middleware Assembly (PRAGMA)
Contact: Sriram Krishnan
Organization(s): San Diego Supercomputing Center (SDSC)
Websites: `http://nbcr.sdsc.edu/`

http://www.pragma-grid.net/
http://avianflugrid.pragma-grid.net/
http://mgltools.scripps.edu/
Tool: [Scripts]/Vision
Description: The Avian Flu Grid project is developing a global infrastructure for the study of Avian Flu as an infectious agent and as a pandemic threat. Figure 11 shows a workflow that is used in drug design. It is used to understand the mechanism of host selectivity and drug resistance.

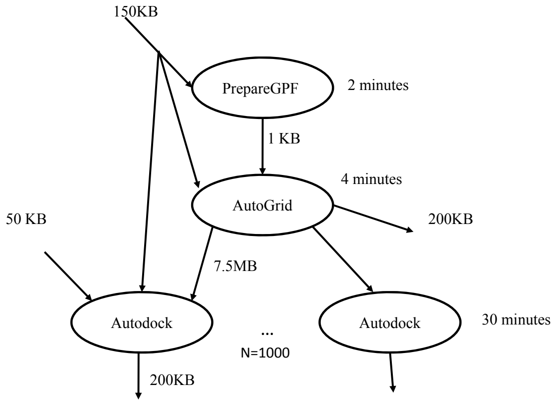

Figure 11: Avian Flu workflow. A workflow used in drug design to study the interaction of drugs with the environment.

3.7 caDSR

Project: cancer Biomedical Informatics Grid (caBIG)
Contact: Ravi Madduri, Wei Tan, Cem Onyuksel
Organization(s): Argonne National Laboratory
Websites: http://www.cagrid.org/
Tool: Taverna
Description: The cancer Biomedical Informatics Grid(caBIG) is a virtual infrastructure that connects scientists with data and tools towards a federated cancer research environment. Figure 12 shows a workflow using the caDSR (Cancer Data Standards Repository) and EVS (Enterprise Vocabulary Services) services [2] to find all the concepts related to a given context. The caDSR service is used to define and manage standardized metadata descriptors for cancer research data. EVS in turn facilitates terminology standardization across the biomedical community.

4 Astronomy and Neutron Science

In this section we consider scientific workflow examples from the astronomy and neutron science community.

4.1 Astronomy workflow

Project: Pan-STARRS
Contact: Yogesh Simmhan
Organization(s): Microsoft Research

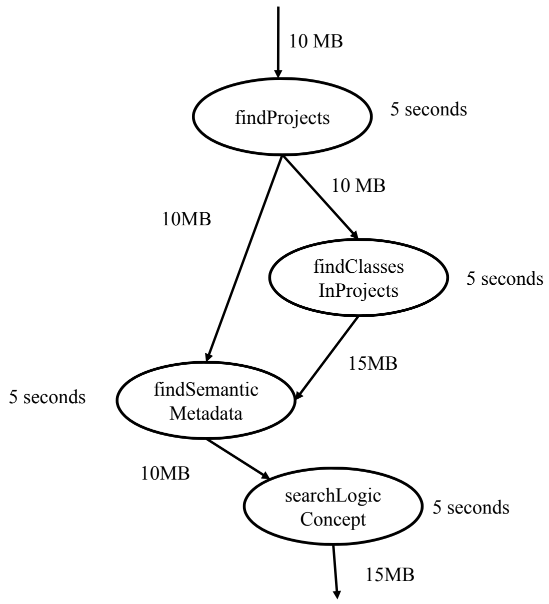

Figure 12: Cancer Data Standards Repository workflow. A workflow used to query concepts related to an input context.

Websites: `http://pan-starrs.ifa.hawaii.edu/public/`
`http://www.ps1sc.org/`
Description: The goal of the Pan-STARRS's (Panoramic Survey Telescope And Rapid Response System) project [19] is a continuous survey of the entire sky. The data collected by the currently deployed prototype telescope 'PS1' will be used to detect hazardous objects in the Solar System, and other astronomical studies including cosmology and Solar System astronomy. The astronomy data from Pan-STARRS is managed by the teams at John Hopkins University and Microsoft Research through two workflows. The first PSLoad workflow (Figure 13) stages incoming data files from the telescope pipeline and loads them into individual relational databases each night. Periodically the online production databases that can be queried by the scientists, are updated with the databases collected over the week by the PSMerge workflow(Figure 14). The infrastructure to support the PS1 telescope data is still under development. Both the Pan-STARRS workflows are data intensive but require coordination and orchestration of resources to ensure reliability and integrity of the data products.

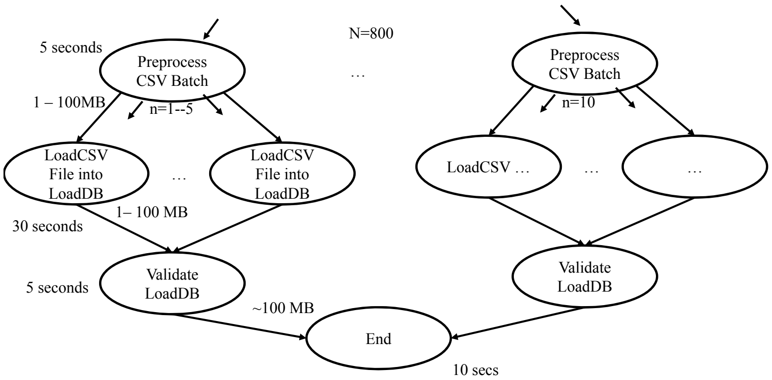

Figure 13: PSLoad workflow. Data arriving from the PS1 telescope is processed and staged in relational databases each night.

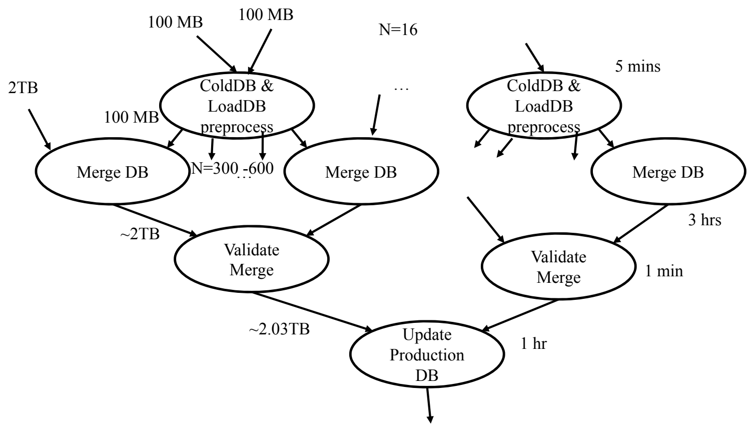

Figure 14: PSMerge workflow. Each week, the production databases that astronomers query are updated with the new data staged during the week.

4.2 McStas workflow

Project: Spallation Neutron Source (SNS), Neutron Science TeraGrid Gateway(NSTG)
Contact: Sudharshan Vazhkudai, Vickie E. Lynch
Organization(s): Oak Ridge National Laboratory
Websites: `http://neutrons.ornl.gov/`
Description: Neutron science research enables study of structure and dynamics of molecules that constitute materials. Neutron Source SNS at Oak Ridge National Laboratory connect large neutron science facilities that contain instruments with computational resources such as the TeraGrid [25]. The Neutron Science TeraGrid Gateway enables virtual neutron scattering experiments. These experiments simulate a beam line and enables experiment planning and experimental analysis. Figure 15 shows a virtual neutron scattering workflow using McStas, VASP, and nMoldyn. VASP and nMoldyn are used for molecular dynamics calculations and McStas is used for neutron ray-trace simulations. The workflow is computationally intensive and currently runs on ORNL supercomputing resources and TeraGrid resources.

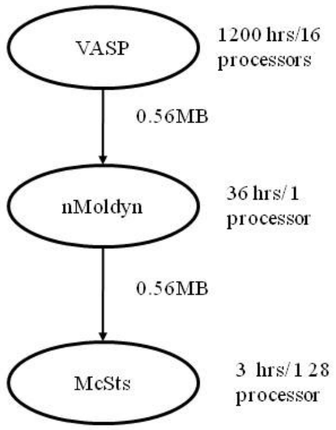

Figure 15: McStats Workflow. This workflow is used for Neutron ray-trace simulations.

5 Computer Science and Business Examples

Workflow tools are increasingly being used in different scenarios both in scientific as well as business processes. In addition programming models like MapReduce [15] facilitate problems to be composed as distinct work units with stated dependencies. In this section we explore some examples that illustrate workflows whose users are often computer scientists or programmers.

5.1 Animation

Rendering computer animation frames is fairly time consuming. Distributed rendering on multiple processors has been known to provide significant speedups over running on a single processor [14]. The animation workflow is based on distributed rendering that is commonly used today for frame generation. The animation workflow has MapReduce style programming model where work is distributed and the results are gathered and synthesized for the final result. The computational and data sizes are rough numbers used for illustration [13, 37].

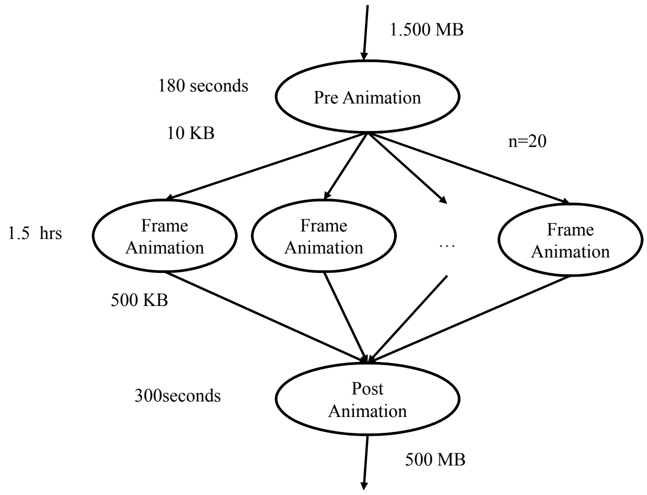

Figure 16: Animation workflow. The rendering work is distributed across a multitude of nodes.

5.2 Performance Measurement Workflow

Applications running in distributed environments like Grid and cloud computing resources often experience significant changes in performance. Benchmarking and performance experiments are often critical in these environments to determine the best binary for a given set of resources. Tilson et al. [34] describe a way to use workflow tools to facilitate the benchmarking of a large number of variable parameters including compiler, link and runtime flags (Figure 17).

5.3 Load balancing as a workflow

Recent computing models have resulted in application middleware investigating mechanisms to dynamically manage the resource pool. Cloud computing services such as Amazon EC2 [1] allow users and applications to increase their resource pool on increased load and decrease the number of resources when the load drops. When

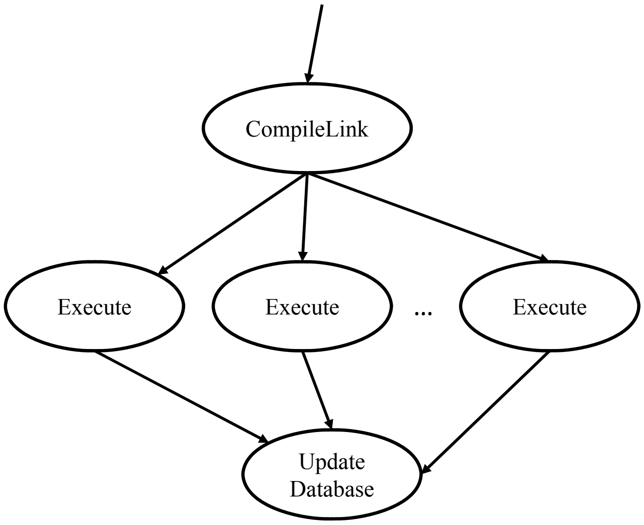

Figure 17: Performance Measurement workflow. The workflow is used for benchmarking applications with various compiler, link and runtime flags.

considering the load from different users or applications that use a defined resource pool we can consider the entire load managed by the middleware to be a "workflow of workflows" where the task dependency might be based on number of concurrent resources available. For example if there are four independent tasks(Figure 18) and just one resource the workflow would be a simple sequential workflow. However if there were two resources available, two tasks would run and then subsequently the remaining two tasks would run. Similarly if three resources were available, three tasks would initially execute in parallel. A similar strategy would be followed for workflows where in addition to the workflow dependencies, virtual dependencies are created between the tasks because of resource requirements. In a more general case consider a cloud computing application that might procure more resources as the load increases and reduce the number of resources as the load decreases. Thus the resources procured or allotted might themselves be represented as a workflow task graph(Figure 19) where each node in the graph represents the resource slot.

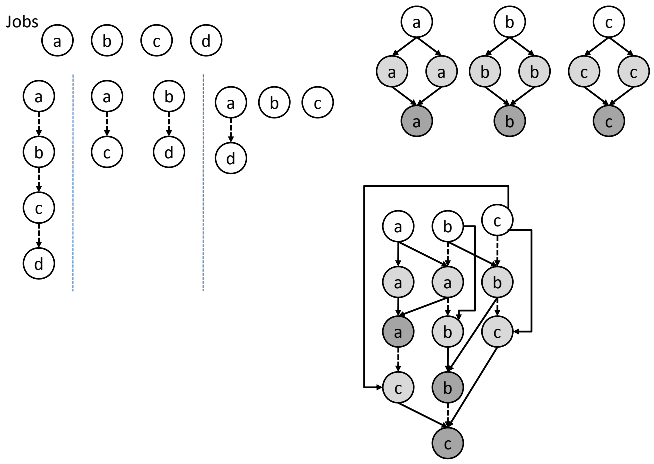

Figure 18: Load balancing workflow. When jobs or workflows are scheduled on resources, a virtual dependency is created from the resource availability constraint that results in a virtual workflow as shown with dotted lines

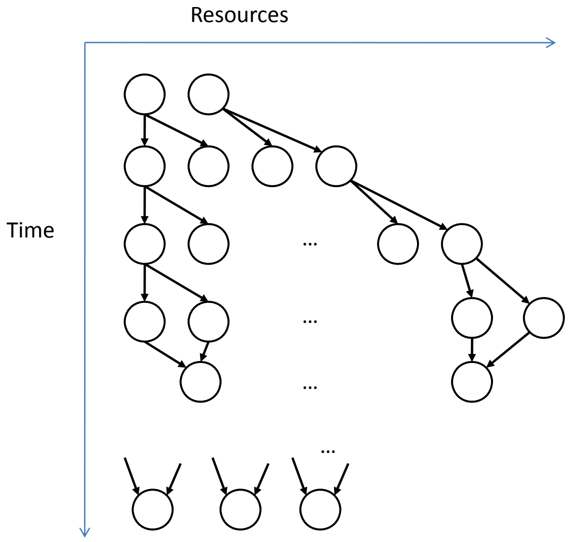

Figure 19: Resource profile as a workflow. A dynamic application manager might procure resources as load increases and release resources as load falls below a threshold. The resource profile over time can be represented as a workflow structure.

6 Discussion

In this paper we have presented a number of workflows from different domains. The workflows have varying requirements and constraints. In this section we provide a higher level discussion on workflow characteristics. Additionally, the workflow examples demonstrate the required support in next-generation workflow and resource management tools to support dynamic and cloud computing environments.

6.1 Workflow Types

The workflows described in this paper vary significantly in their computational and data requirements. A number of the bioinformatics workflows often have tasks that are based on querying large databases in order of minutes for the task execution. In other cases we see each of the tasks of the workflow require computation time on the order of hours on multiple processors. In some cases sub-parts of the workflow might also present different characteristics. In addition, the sizes of the intermediate data products might also vary. Workflow management strategies for each of these workflows can vary and thus require the understanding of the workflow to apply appropriate techniques. In this section we consider the characteristics that help classify the workflow types that are observed. We also present the challenges that each of the workflow types present.

Structure. The size of the workflow is an important characteristic to determine resource requirements, etc. We consider the number of tasks of the workflow as its structural characteristic. The size of the workflows that are deployed today in most production environments are relatively small. The largest workflows in our set contain about a couple of hundred independent tasks. The Avian Flu workflow(Figure 11) has about a thousand nodes but the computation at each node is expected to take a few minutes. Scientists express a need to run larger sized workflows but are often limited by available resources or workflow tool features that might be needed to support such large-scale workflows. It is also often difficult in grid environments today to scale workloads up or down due to batch queue wait times and other factors. In addition to the total number of tasks in a workflow it is also important to consider the width and length of the workflows. The width of the workflow (i.e. maximum number of parallel branches) determines the concurrency possible and the length of the workflow characterizes the makespan (or turnaround time) of the workflow. We observe that in our workflow examples, the larger sized workflows such as the Motif workflow (Figure 8) and the Pan-STARRS workflows (Figures 13 and 14) the width of the workflow is significantly larger than the length of the workflow.

Pattern. The workflows that we surveyed depict the basic control flow patterns such as sequence, parallel split, synchronization [35]. The parallel split-synchronization pattern has similarities to the MapReduce programming paradigm [15]. A number of workflows divide the work units into manageable part and the results are then combined - e.g. Animation (Figure 16), Motif workflow (Figure 8), Pan-STARRS workflows (Figures 13 and 14).

Computation. In addition to the structure and pattern of a workflow it is important to understand the computational requirements. In the presented workflow examples we observe that computational time required by the workflows can vary from a few seconds to several hours. A number of the bioinformatics workflows depend on querying large databases and have small compute times. Some examples include the Glimmer workflow (Figure 6), Gene2Life (Figure 7), caDSR (Figure 12). Similarly the initial parts of the LEAD forecast workflow(Figures 1 and 2) and the LEAD data mining workflows (Figure 3) have small computational load. A number of the workflows including the forecasting parts of the LEAD workflow, Pan-STARRS production merge workflows (Figure 14), SCOOP (Figure 4), SNS (Figure 15), Motif (Figure 8), NCFS (Figure 5) have medium to large sized compute requirements.

Data. The workflows are associated with different types of data including input data, backend databases, intermediate data products, output data products. A large number of the bioinformatics applications often have small input and small data products but often rely on huge backend databases that are queried as part of task execution. These workflows require that the databases be pre-installed on various sites and resource selection is often based on selecting the resources where the data might be available. Workflows such as LEAD (Figures 1 and 2), SCOOP (Figure 4), NCFS (Figure 5) have fairly large sized input, intermediate and output data products. The Glimmer workflow (Figure 6) has similar sized input and output data products but its intermediate data products are smaller. In today's production environments workflows often compress data products to reduce transfer times through intermediate scripts etc. When scheduling workflows on resources, a number of data issues need to be considered including the availability of the required data as well as the data transfer time of both input and output products.

The combination of the structural and pattern characteristics, the computational and data sizes helps in understanding the workflow requirements when making planning and adaptation decisions.

6.2 Multiple workflows

The user interacts with applications through various portal and graphical interfaces for workflow tools. Workflow management techniques today are focused on managing single workflows in a distributed environment like the grid [26, 36]. However portal environments facilitate multi-user simultaneous access to the same workflows and underlying resources. In addition, a number of scientific explorations including the weather and ocean modeling (Section 2) often require a large number of parallel runs to be launched to study different parameters to increase result accuracy.

Competing workflows. Portal and gateway environments allow a number of workflows from different users to be launched simultaneously. In such cases workflows from different users are often competing for the same resources. We discussed the LEAD scenario where a weather forecasting workflow will need to have higher priority than a workflow launched by a user in an educational workshop. Workflow management techniques needs to account for the different classes of workflow users when allocating resources.

Cooperating workflows. Scientists often conduct parametric or exploratory studies that involve launching multiple parallel workflows. In addition, there might be workflows from different users which have the same priority and similar constraints requiring them to be managed to ensure fairness. There is limited capabilities to be able to ensure such policies in the workflow engines available today.

Data sharing and reuse. When mutliple workflows exist in the system, there is an opportunity to save computational time by reusing data products from identical executions [16]. However in these situations it is also important to manage data privacy concerns when managing data products from potentially competing workflows.

Thus we need tools and mechanisms to manage competing and cooperating workflows in a system. The multiple workflow scenario can be composed using the MapReduce [15] programming model. Each of the workflows will correspond to a map operation and the merging of the results from each workflow will be the reduce operations. Additional research into using tools like Hadoop [3] for managing multiple workflows is required.

6.3 Workflow Capabilities

Workflow tools have limited capabilities today to allow users to specify constraints and other expectations from their workflows. We investigate some such constraints that users might need to express in conjunction with workflow descriptions through workflow composition tools.

Exploratory. Scientific explorations often have uncertainties that might need to be resolved during runtime. Input data sizes can vary largely affecting the characteristics of the workflow. In a number of explorations scientists and their workflows interact with real-time data collecting instruments such as the Large Hadron Collider (LHC) [4], sensors, radars [18, 27], etc. Thus in some cases while a general structure of the workflow might be known, the exact characteristics of the workflow is determined during execution.

Interactive. Business workflows and scientific explorations often require a "human-in-the-loop" as part of the workflow. Workflow management techniques often have to consider sub-parts of the workflow for scheduling and adaptation.

Constraints. In addition to the workflow description, users often need to specify various constraints on the workflow. The weather and ocean modeling workflows (Section 2) are time-sensitive. The workflow results must be obtained in advance for weather response agencies to be take appropriate action. In addition the cost of resources (either allocation seconds on TeraGrid or real dollars on resources such as Amazon EC2) might be a consideration for the end user.

6.4 Resource coordination.

Scientific workflows largely run in batch queue based grid environments and business workflows run on monolithic corporate systems. However the advent of utility and cloud computing systems can change the interaction mode. Cloud computing systems allows users to customize software environments allowing workflow tools to be able to manage application specific software and data on the resources. In addition procuring resources in advance for later workflow steps can be achieved with the new resource access mechanisms thus minimizing workflow makespan by reducing resource wait times. Thus new mechanisms are required in workflow and resource management tools.

7 Summary

In this paper we have investigated workflows from various domains that have different structures, computational and data requirements. Understanding the characteristics of the workflows and other capabilities and constraints desired from the workflow is necessary for applying specific orchestration techniques. In addition, enabling additional operations such as MapReduce, file compression and decompression as standard operations available in workflow tools needs to be investigated. Runtime management of workflow instances will also be necessary as we move to more complex workflows with uncertainties and dynamic resource environments such as cloud computing.

References

[1] Amazon web services. http://aws.amazon.com/.

[2] Cagrid taverna workflows. http://www.cagrid.org/wiki/CaGrid:How-To:Create_CaGrid_Workflow_Using_Taverna.

[3] Hadoop. http://hadoop.apache.org/core.

[4] Large hydron collider. http://lhc.web.cern.ch/.

[5] North Carolina Floodplain Mapping Program. http://www.ncfloodmaps.com/.

[6] Renci computational resources. http://www.renci.org/resources/computing.php.

[7] Scoop website. http://scoop.sura.org.

[8] Unidata Local Data Manager(ldm). http://www.unidata.ucar.edu/software/ldm/.

[9] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludscher, and S. Mock. Kepler: An extensible system for design and execution of scientific workflows, 2004.

[10] S. F. Altschul, W. Gish, E. M. W. Miller, and D. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 214(1-8), 1990.

[11] D. Atkins. A report from the U.S. national science foundation blue ribbon panel on cyberinfrastructure. In *CCGRID '02: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, page 16, Washington, DC, USA, 2002. IEEE Computer Society.

[12] B. Blanton, H. Lander, R. A. Luettich, M. Reed, K. Gamiel, and K. Galluppi. Computational Aspects of Storm Surge Simulation. 2008.

[13] A. Chong, A. Sourin, and K. Levinski. Grid-based computer animation rendering. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 39–47, New York, NY, USA, 2006. ACM.

[14] T. W. Crockett. An Introduction to Parallel Rendering. *Parallel Computing*, 23(7):819–843, 1997.

[15] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. pages 137–150.

[16] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, K. Blackburn, A. Lazzarini, A. Arbree, R. Cavanaugh, and S. Koranda. Mapping abstract complex workflows onto grid environments. *Journal of Grid Computing*, 1:25–39, 2003.

[17] E. Deelman and Y. Gil. Report from the NSF workshop on the challenges of scientific workflows. *Workflow Workshop*, 2006.

[18] K. K. Droegemeier, D. Gannon, D. Reed, B. Plale, J. Alameda, T. Baltzer, K. Brewster, R. Clark, B. Domenico, S. Graves, E. Joseph, D. Murray, R. Ramachandran, M. Ramamurthy, L. Ramakrishnan, J. A. Rushing, D. Weber, R. Wilhelmson, A. Wilson, M. Xue, and S. Yalda. Service-Oriented Environments for Dynamically Interacting with Mesoscale Weather. *Computing in Science and Engg.*, 7(6):12–29, 2005.

[19] N. K. et al. Pan-STARRS collaboration. *American Astronomical Society Meeting*, (206), 2005.

[20] W. Kramer and C. Ryan. Performance Variability of Highly Parallel Architectures. International Conference on Computational Science, 2003.

[21] H. M. Lander, R. J. Fowler, L. Ramakrishnan, and S. R. Thorpe. Stateful grid resource selection for related asynchronous tasks. Technical Report TR-08-02, RENCI, North Carolina, April 2008.

[22] X. Li, B. Plale, N. Vijayakumar, R. Ramachandran, S. Graves, and H. Conover. Real-time storm detection and weather forecast activation through data mining and events processing. *Earth Science Informatics*, May 2008.

[23] B. Ludscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system, 2005.

[24] R. Luettich, J. J. Westerink, and N. W.Scheffner. ADCIRC: An advanced threedimensional circulation model for shelves, coasts and estuaries; Report 1: theory and methodology of ADCIRC- 2DDI and ADCIRC-3DL. *Technical Report DRP-92-6, Coastal Engineering Research Center, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS*, 1992.

[25] V. Lynch, J. Cobb, E. Farhi, S. Miller, and M. Taylor. Virtual Experiments on the Neutron Science Teragrid Gateway. *TeraGrid*, 2008.

[26] A. Mandal, K. Kennedy, C. Koelbel, G. Marin, J. Mellor-Crummey, B. Liu, and L. Johnsson. Scheduling strategies for mapping application workflows onto the grid. In *High Performance Distributed Computing (HPDC 2005).*, pages 125–134. IEEE Computer Society Press, 2005.

[27] B. Plale, D. Gannon, J. Brotzge, K. K. Droegemeier, J. Kurose, D. McLaughlin, R. wilhelmson, S. Graves, M. Ramamurthy, R. D. Clark, S. Yalda, D. A. Reed, E. Joseph, and V. Chandrashekar. CASA and LEAD: Adaptive Cyberinfrastructure for Real-time Multiscale Weather Forecasting. *IEEE Computer*, (39):66–74, 2006.

[28] L. Ramakrishnan, B. O. Blanton, H. M. Lander, R. A. Luettich, Jr, D. A. Reed, and S. R. Thorpe. Real-time Storm Surge Ensemble Modeling in a Grid Environment. In *Second International Workshop on Grid Computing Environments (GCE), Held in conjunction ACM/IEEE Conference for High Performance Computing, Networking, Storage and Analysis*, November 2006.

[29] L. Ramakrishnan, M. S. Reed, J. L. Tilson, and D. A. Reed. Grid Portals for Bioinformatics. In *Second International Workshop on Grid Computing Environments (GCE), Held in conjunction with ACM/IEEE Conference for High Performance Computing, Networking, Storage and Analysis*, November 2006.

[30] L. Ramakrishnan, Y. Simmhan, and B. Plale. Realization of Dynamically Adaptive Weather Analysis and Forecasting in LEAD. In *In Dynamic Data Driven Applications Systems Workshop (DDDAS) in conjunction with ICCS (Invited)*, 2007.

[31] I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields. *Workflows for e-Science: Scientific Workflows for Grids*. Springer, December 2006.

[32] J. Tilson, A. Blatecky, G. Rendon, G. Mao-Feng, and E. Jakobsson. Genome-Wide Domain Analysis using Grid-enabled Flows. 2007.

[33] J. Tilson, G. Rendon, G. Mao-Feng, and E. Jakobsson. Motifnetwork: A Grid-enabled Workflow for High-throughput Domain Analysis of Biological Sequences:Implications for study of phylogeny, protein interactions, and intraspecies variation. 2007.

[34] J. L. Tilson, M. S. Reed, and R. J. Fowler. Workflow for Performance Evaluation and Tuning. *IEEE Cluster*, 2008.

[35] W. M. P. van der Aalst, Ter, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, July 2003.

[36] Y.Zhang, A. Mandal, H.Casanova, A. Chien, Y. Kee, K. Kennedy, and C. Koelbel. Scalable Grid Application Scheduling via Decoupled Resource Selection and Scheduling. CCGrid, May 2006.

[37] Y. Zhou, T. Kelly, J. L. Wiener, and E. Anderson. An extended evaluation of two-phase scheduling methods for animation rendering. In *Job Scheduling Strategies for Parallel Processing, 11th International Workshop*, pages 123–145, 2005.