Design of a Grid workflow for a climate application

Joerg Schneider, Julius Gehr and Hans-Ulrich Heiss
Technische Universitaet Berlin, Germany
{komm, jules, heiss}@cs-tu-berlin.de

Tiago Ferreto and César De Rose
Pontifcia Universidade Catlica do Rio Grande do Sul, Brazil
{tiago.ferreto, cesar.derose}@pucrs.br

Rodrigo Righi, Eduardo R. Rodrigues, Nicolas Maillard and Philippe Navaux
Universidade Federal do Rio Grande do Sul, Brazil
{rodrigo.righi, eduardo.rocha, nicolas, navaux}@inf.ufrgs.br

Abstract

Grid applications can be modeled as a composition of rather independent tasks. There are two approaches to define such a workflow either by combining multiple applications to build a more complex functionality or by splitting up an existing application. In this paper we analyze the latter process. We present a compute intensive application for climatology simulation and the options available to split it up. Using the simulation mode of our Grid broker, we were able to compare the different workflow specifications before actually executing the workflows. This case study showed, using finer grained workflows–which usually need more adjustments to the software–allows better performance in the Grid.

1. Introduction

Grid technology enables users located at different sites to get a unified access to various resources even across organizational borders. By aggregating a number of compute resources, Grid technology can be used to execute complex applications and deliver their results in less time. Additionally, the Grid can also manage other types of resources reaching from computers, storage space, and network capacity up to access to databases or scientific instruments.

In order to map complex applications on multiple resources, the applications are modeled as a composition of functional components that assemble a workflow [9]. Usually, each component of the workflow defines a part of the application to be executed on a single resource, which could also be a parallel computer. Furthermore, the workflow defines all dependencies between these components, e.g., output files needed as input for the following steps. Workflows can be used to couple applications, like using some simulation software together with a 3D renderer or some statistic analysis toolbox. In this case, the components are already defined and the dependencies are naturally given by the composition.

Porting a single existing application into a Grid workflow means to identify parts, which are rather independent, and the dependencies between them. There are usually multiple Grid workflow specifications for the same application, resulting in different performance in the Grid and different effort for porting and introducing additional communication interfaces.

Concerning this, simulation is an attractive approach to evaluate a workflow before actually executing it on a real Grid or starting to port the application. Virtual Resource Manager (VRM) [1, 3] is a Grid broker that features a mode which allows simulating the scheduling and execution of Grid workflows in a Grid. The Grid model can be derived from a real world installation or a synthetic setup.

In this work, we used the VRM simulation mode to evaluate two different workflow specifications that compute the climate for a specific region in South Brazil. The meteorological forecast model chosen is the BRAMS which includes parameterizations for tropical regions [20]. We profiled this application, identified the key components and organized it as a Grid workflow. The Grid model used reflects the organization and the features of our transatlantic Grid that joins resources located in Brazil and Germany. The simulation evaluation indicates the best strategy to be used to actually run the climatology application on the real Grid avoid the cost of performing real executions. This paper de-

scribes the climatologic application, our workflow and Grid models and experimental results.

2. Climatology

Numerical weather forecasts predict the atmospheric behavior by using physical models. These models are comprised of a set of partial differential equations describing the fluid dynamical behavior of the atmosphere. Applying a time integration scheme on these differential equations, the state of the atmosphere in the future is determined based on an initial state. Typically, this procedure is performed in a series of steps, each one advancing the state of the atmosphere representation by a fixed amount of time.

However, the time integration scheme cannot be applied for a long period of time. As Lorenz [14] observed, the time integration of the atmospheric model amplifies any incompleteness of initial condition as well as the imperfections of the model. Thus, a day-to-day forecast is useful only for a few days.

The atmospheric behavior in some regions is much more influenced by other factors. For instance, in [18] is pointed out that tropical flow patterns and rainfall are strongly determined by the sea-surface temperature underlying a region. Thus, for these regions it is possible to perform longer term forecasts.

This observation provides the scientific basis for long-term forecasts. But, even though it is possible to execute the model for a long period, usually systematic tendency in the model lead to biased results. In order to minimize this bias, a subtraction operation is taken between the forecast itself and a monthly average of the forecast over a set of years in the past. This set of averages is called the climate normal and the forecast is presented as the subtracting result, called anomaly [10].

The amount of work needed to produce the climate normal depends basically on how many years are averaged, the spatial dimensions of the forecast and how many ensembles (members) are averaged. A common practice is to obtain the average for up to 30 consecutive years [16], that represents a very long execution of the model. The dimension of the simulation is determined by the employed model. A much known model classification is based upon its horizontal domain - global models (entire world) and regional models (a limited area of the world). Besides the monthly average, the climate normal typically includes averages among some ensembles, which are the execution of the model with different initial conditions. This is done to improve the representativeness of the results.

In this work, we consider the climate normal production for a region in Brazil using the BRAMS model [20]. BRAMS means Brazilian Regional Meteorological System and is a model derived from RAMS that includes tropical parameterizations. This model receives as input a lower resolution data from another meteorological model and produces a higher resolution output for a specific region. Since the climatology is a very time consuming activity and involves large amounts of data, grid computing can be applied to produce the climate normal in less time.

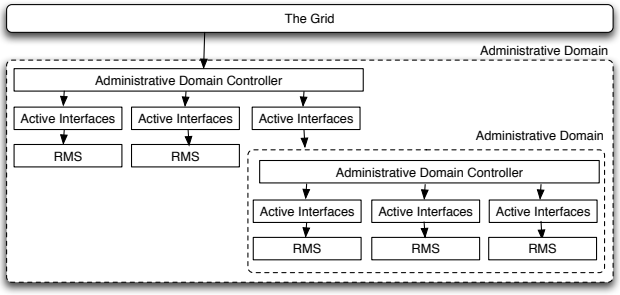


Figure 1. Hierarchical Administrative Domain Structure

3. Virtual Resource Manager - VRM

The Grid setup in this work is managed by the *Virtual Resource Manager* (VRM) [1]. The VRM architecture focuses on support for QoS by service level agreements (SLA). Therefore, the allocation and admission of jobs have to be guaranteed right at the submission time.

Using advance reservation in contrast to the classical queuing approach, for each job the actual start – and therefore also end time – as well as resource allocation is fixed at submission time. Later arriving reservations are planned to not overlap with the already admitted. Advance reservation is used in the Grid broker and within the local resource management systems. Nonetheless, the VRM is also capable to integrate a resource manager without advance reservation [2].

Figure 1 shows the architecture of the VRM. The *Administrative Domain Controller* (ADC) constitutes the central management component of the VRM architecture. The ADC is responsible for establishing the *Administrative Domains* (AD) which consist of a number of underlying local resources and their local *Resource Management Systems* (RMS). These management systems may control arbitrary types of resources, e.g., cluster systems, parallel computers or networks and are connected to the ADC by *Active Interfaces*.

Using elaborated reservation protocols [17] and scheduling algorithms [7, 4], VRM is able to provide an SLA without requiring the local resource provider to give up their autonomy [2]. By sending only specific requests to the resources, the central ADC will never get a complete view

on the local schedules. Therefore, the actual utilization and usage profile of the participating sites is hidden.

3.1. Simulation Mode

VRM usually serves as a Grid management system and is itself object to investigations of performance behavior in real world configurations using test bed installations - like the Brazilian-German Grid analyzed in this paper. The advantage of our VRM Grid management infrastructure, on the other hand, is the ability to be also use in a simulation-based environment. In the simulation mode, the infrastructure is steered by a discrete event framework, but uses the same code base as in the real VRM mode.

By running the VRM in simulation mode the Grid management system is able to support a large spectrum of different configurations from simple organization-based installations to worldwide Grid scenarios. In addition, the VRM Grid management infrastructure is able to integrate simulated resources as well as real resources. Such kind of mixture between simulation and real world instances can easily implement huge Grid infrastructures by simulating lots of resources and local resource management systems as well as the associated active interfaces. This simulation environment can be connected to resources, such as cluster systems building the existing test bed. On the other hand the simulation of the VRM framework on the whole can overcome limitations coming with real world Grid instances, such as limited amount of resources and long runtime, as the simulation implemented as discrete event simulation provide time reduction in case no simulation events are present. The environment used to investigate the performance of the novel Grid reservation protocol is based on our VRM framework using the simulation mode.

3.2. Grid workflows

VRM supports Grid workflows, i.e., the reservation can consist of any number of sub-tasks. Giving just a general deadline and the dependencies between the sub-tasks, the user gives the Grid scheduler the freedom to place the sub-tasks on any matching resource. Dependencies usually arise, if a task needs the output of another task as input. In order to not delay the actual start of each task, the user can provide additional information about the data size. VRM will take care to schedule the succeeding tasks late enough to have all input data ready. If a managed network is available, it even reserve the necessary bandwidth to guarantee this.

Grid workflows enable the user to define specific points within one's application, where the application can be split on multiple resources. On the other hand, the user can combine multiple applications he otherwise would start manually in a specific order into a single workflow. Both applications of the Grid workflow concept result in a more efficient usage of the Grid.

4. Related Work

Advance reservation is an important allocation strategy that provides simple means for reliable planning and co-allocation of heterogeneous resources. Besides flexible support for co-allocations, advance reservations also have other advantages such as an increased admission probability when reserving sufficiently early and reliable planning for users and operators. In contrast to the independent usage of several different resources, where also queuing approaches are conceivable, advance reservations have a particular advantage when time-dependent co-allocation is necessary. Advance reservation support has been proposed for several management systems for distributed and parallel computing [1, 11, 19]. In [1], advance reservations have been identified as being essential for a number of higher level services such as the support of SLA.

Complex applications requiring multiple resources are becoming a major application of the Grid [15]. Such Grid workflows increase the complexity of the allocation process and efficient scheduling and allocation schemes have to be developed.

To declare the composition of workflows a number of description languages have been proposed. Some of them are based on well known languages for behavior modeling, like petri nets, e.g., used in the Fraunhofer Resource Grid [12]. Another approach is used for the Grid service flow language (GSFL) by adopting concepts from the web service composition domain [13].

The GRAAP working group of the Global Grid Forum (GGF) works on a specification for modeling of service level agreements [6]. The specification does not provide any means to describe Grid workflows, but it introduces techniques to define multiple Grid jobs in a single agreement, which can be extended to a workflow description.

The architectures proposed for Grid workflow handling are usually composed of a user tool and the workflow execution engine. In order to specify the workflow, the user tool composes the sub-tasks. The user tool also calls the execution engine which controls the execution of the workflow within the Grid [21, 12, 5]. In some architectures there are additional layers to enhance the workflows, e.g., by splitting up abstract tasks into concrete sub-workflows [12].

The workflow execution engine is usually realized as a central instance that interacts within the Grid. In some cases the used Grid resources exchange the input and output data directly, but even then there is an additional central instance coordinating the control flow [13]. In the GridFlow architecture [5] multiple execution engines are used, but the con-

trol flow of a workflow is always handled by the same system.

While the development of a workflow-enabled Grid scheduler is a current research topic, details about how scientific workflows will actually look like are mostly unclear. Deelmann et.al. [8] describes a number of Grid workflow setups from astronomy, biology and physics. Because of the number of examples, the descriptions are rather briefly and don't show the exact profile of the referred applications. The formerly mentioned publications on workflow scheduling usually also include a briefly described example of a Grid workflow.

5. BRAMS Application and Workflow

In this section we follow the steps to port the BRAMS application into a Grid workflow. First we analyze the application and identify its basic building blocks. Based on this, we propose two alternative Grid workflow models representing the BRAMS application.

5.1. BRAMS Application

BRAMS [20] is a meteorological model software widely used in Brazil for weather and climate forecasts. As it uses a regional model approach, first a particular region has to be selected for simulation. For the selected region, a set of regularly spaced points is defined to form a grid overlay. Each of these points is then used as the discretized representation of the atmosphere in its surrounding. Increasing the number of points leads to a smaller area represented by each point and thus to a higher resolution. Using multiple nested overlay grids at the same time helps to get detailed results for a sub region.

Basically, within an execution of BRAMS four phases can be identified:

- The first phase is the definition of surface characteristics for all points of the grid overlay based on some more general model of the region. For each selected grid overlay a surface file is created.

- In the second phase an objective analysis is performed. This analysis results in variable initialization files. The number of files depends on the duration of the forecast and the selected grid overlay.

- The core of the simulation is the third phase. In this phase a part of the time is simulated, e.g., a month. The simulation produces the forecast for the specified period. Based on this forecast the new state of the soil and atmosphere model can be derived, which is used as the input to simulate the next period. Statistics about the simulated time period like the amount of rain can also be derived from this forecast.

  This phase is the most compute intensive. It can be executed in parallel in an MPI environment or sequentially in a single machine.

- The fourth phase is for data post processing. It receives data from the third phase and produces visual and textual summaries for each simulated time period, which than will be analyzed by the climatologists.

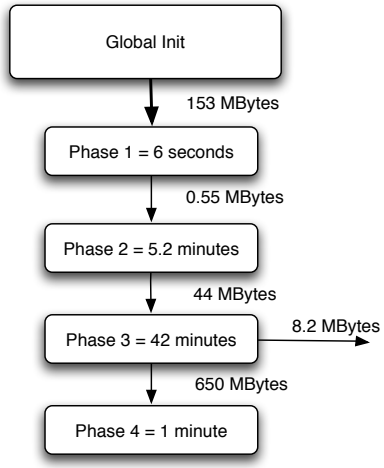


Figure 2. Computational time, input and output data for each BRAMS Phase

These four phases are executed in both weather and climate forecasts. However, the climate forecast requires a much longer execution. In this work, we are considering two nested grid overlays over South Brazil. The first one has $30 \times 28 \times 27$ grid points and 160 km of horizontal resolution. In the inner region a finer second grid overlay with $30 \times 30 \times 27$ points and 40 km of horizontal resolution is used.

Figure 2 gives an overview on the execution times and amount of data transfered between phases for a single 1 month forecast. For a climatologic analysis, a number of months – up to 30 years – is simulated, each requiring an execution of each of these four phases. Furthermore, in climatology multiple forecasts with varying input values and model parameters are computed and compared. Therefore, the required resources can exceed the resources available in many forecast centers. A solution for this problem is spreading the execution in the Grid.

5.2. Workflows

To distribute the computational work, a simple approach would just take each forecast as an atomic unit. But this

would mean to get a cooperation partner in the Grid to spend its resources for a long time only to execute this forecast and maybe delaying local jobs.

In order to get a higher acceptance, the work must be split in smaller units. As written before, one way to get a Grid workflow is to combine existing applications such that they generate a higher level output without further interaction and the other is to split a long running application in smaller pieces.

The description in the previous section with the well defined phases is the result of the first step, i.e., identifying the core working units within the application. In the second step a Grid workflow designer has to identify the parts which can be split or which should be better executed on the same site, e.g., due to communication constraints. Some of the parts, which have been identified as rather independent from the logic view, share a lot of internal data structures and program code and can only be split with a lot of additional programming work or even unnecessary recalculations during runtime due to missing data.

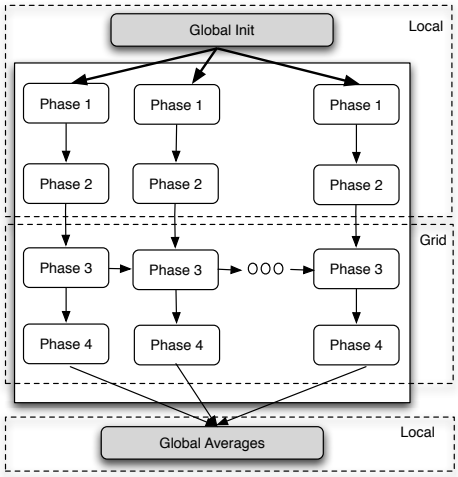
In our case with the BRAMS application, the third phase is already provided as a separate application, either as an MPI application or a single threaded one. A straightforward splitting is to execute all preparing steps (phases 1 and 2) on the user's site, submit a job for simulating each month to the Grid (phase 3) and collect all results for post processing (phase 4) back at the user's site. Figure 3 shows a workflow based on this concept.

This approach is often employed to make use of the Grid. The benefit for the user is that only minor adjustments to the program are necessary. However, a lot of processing is required on the user's site.

As Figure 2 shows, the amount of data forwarded from the first phase to the second is rather big, which, due to networking constraints usually found in grid environments, indicates that this workflow should provide the best performance. Nevertheless, we modeled a second workflow, where all phases are treated as a single job and therefore may be executed on any site in the Grid. This workflow is shown in Figure 4. The main advantage of this workflow is the decentralizing approach, since each Phase 1 and Phase 2 can be assigned to any Grid resource - including the local resources made available in the Grid. Although this idea avoids a processing bottleneck, it has a drawback of much more network communication.

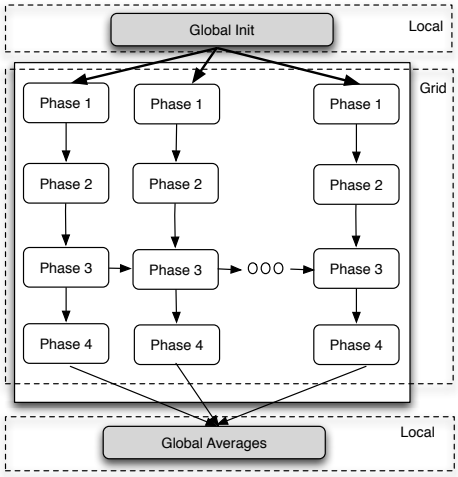
6. Simulation and Results

In this work we used VRM to evaluate the two workflow approaches in order to identify the most efficient one. We defined a model of the Brazilian-German Grid our groups maintain and using this model we were able to simulate multiple months of Grid usage using a variable workload. In



(a) Workflow 1

Figure 3. Workflow 1: The first and second phase as well as the last phase is computed at the submitters site and only the compute intensive third phase is executed in the Grid.



(b) Workflow 2

Figure 4. Workflow 2: All parts can be executed on any Grid node including the ones at the submitters site.

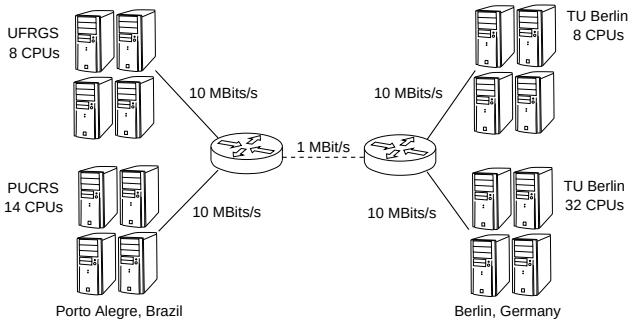


Figure 5. The German-Brazilian Grid setup which was the base for the simulations.

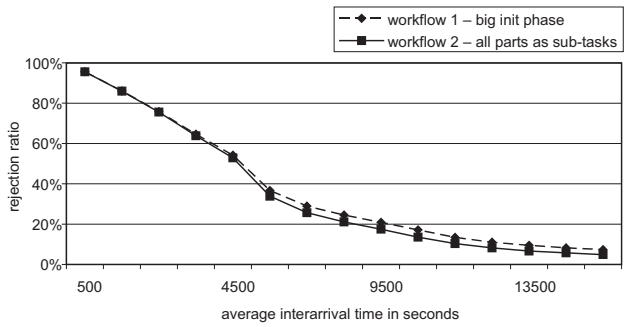


Figure 6. The rejection ratio for different load situations. A low interarrival time yields to a higher loaded Grid.

this section we give an overview of our Brazilian-German Grid architecture, describe how we set up the simulation and obtained the results.

6.1. Brazilian-German Grid

In order to experience the problems of an intercontinental Grid setup, our groups set up a small experimental Grid. The Grid consists of 4 sites. Each site belongs to a different organization and has a different local resource management system.

Although the Grid resource management we use (VRM) is also capable of managing network resources between the sites, the sites are connect to the internet on a best-effort basis. In order to enhance the scheduling decision, VRM uses a virtual network with the measured average bandwidth and latencies between sites. Because VRM uses advance reservations to schedule the workflow in the Grid, it may rather use a resource closer to the data producer that will take a short time to be available than a resource on the other side of the Atlantic which is available right away.

Figure 5 shows the grid setup. The network link between the sites within the same city was measured to provide at least 10 Mbit/s, while the intercontinental link only provides 1 Mbit/s. In the simulation all sites were assumed to have homogenous processors and thus the same execution times for all jobs. This abstraction does not constitute a restriction, since the focus of the simulation is to compare workflows with different communication patterns. Heterogeneous computing resources only lead to a more complex calculation of the execution time, and will trigger the same decisions by the Grid resource management system.

6.2. Simulation and Evaluation Criteria

In order to simulate the usage of the Brazilian-German Grid by climatologists, 1000 workflows for each BRAMS workflow model were generated. Each one of these executions corresponds to a single forecast setup. The size of the executions varied between 12 month and 30 years forecasts - resulting in execution times between hours and 8 days. The interarrival time between the workflows was varied to simulate different load situations. With a short interarrival time, a large number of workflows have to be scheduled for the same time frame, generating a high load.

VRM guarantees the availability of sufficient resources during the admission of a workflow. Therefore, if it can't determine and reserve a possible schedule for a submitted workflow, it will be rejected. The general goal is to reject as less workflows as possible. The number of rejections per submitted workflows (rejection ratio) is used to evaluate the quality of the schedules and consequently the user experience.

6.3. Results

Using the simulation mode of VRM we were able to simulate months of Grid usage in different load situations.

As Figure 6 shows, if each part of the application is modeled within the workflow on its own (Workflow 2), the rejection ratio is reduced up to 3 percent points. Even in situations with a high load, VRM can use the fine-grained definition of the jobs to determine a valid schedule by placing all parts on different sites, accepting more and longer lasting file transfers.

7. Conclusion

Porting a scientific application in a Grid workflow is not an easy task. There are a couple of decisions to be taken, depending on the expected programming work and the gain of splitting the application. Using a generated workload of workflows and simulating their scheduling in a modeled

Grid can help the Grid workflow designer before porting the application and without long runtimes.

By using VRM in simulation mode we were able to simulate the Brazilian-German Grid test bed and test which workflow description suits best for this setup. We used a widely used climatology application with different Grid workflow descriptions. The simulations showed that giving the scheduler more freedom by composing the workflow of fine grained steps - even if they are likely to be computed at the same site - results in a lower rejection rate. This result helped us find the best available workflow to execute the BRAMS application on the real Grid.

Since VRM is a full featured Grid resource management system and will be used to manage the Brazilian-German Grid test bed, we strongly believe that the simulation results presented in this paper will be reproduced in the real Grid test bed. As future work we intend to reproduce our simulations in a real Grid and investigate how to further assist the workflow generation process for Grid applications.

References

[1] L. Burchard, M. Hovestadt, O. Kao, A. Keller, and B. Linnert. The virtual resource manager: an architecture for SLA-aware resource management. *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pages 126–133, 2004.

[2] L.-O. Burchard, H.-U. Heiss, B. Linnert, J. Schneider, O. Kao, M. Hovestadt, F. Heine, and A. Keller. The virtual resource manager: Local autonomy versus QoS guarantees for grid applications. In V. Getov, D. Laforenza, and A. Reinefeld, editors, *Future Generation Grids*, volume 2 of *CoreGrid*, 2006.

[3] L.-O. Burchard, C. A. F. D. Rose, H.-U. Heiss, B. Linnert, and J. Schneider. VRM: A failure-aware grid resource management system. In *SBAC-PAD '05: Proceedings of the 17th International Symposium on Computer Architecture on High Performance Computing*, pages 218–227. IEEE Computer Society, 2005.

[4] L.-O. Burchard, C. A. F. D. Rose, H.-U. Heiss, B. Linnert, and J. Schneider. VRM: A failure-aware grid resource management system. In *Proceedings of the 17th International Symposium on Computer Architecture and High Performance Computing*, pages 218–225. IEEE press, Oct. 2005.

[5] Cao, J., S. Jarvis, S. Saini, and G. Nudd. GridFlow: Workflow Management for Grid Computing. In *3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003), Tokyo, Japan*, pages 198–205, May 2003.

[6] Czajkowski, K., A. Dan, J. Rofrano, S. Tuecke, and M. Xu. Agreement-based Service Management (WS-Agreement, Draft). https://forge.gridforum.org/projects/graap-wg, 2004.

[7] J. Decker and J. Schneider. Heuristic scheduling of grid workflows supporting co-allocation and advance reservation. In B. Schulz, R. Buyya, P. Navaux, W. Cirne, and V. Rebello, editors, *7th Intl. IEEE Intl. Symposium on Cluster Computing and the Grid (CCGrid07)*, pages 335–342, Rio de Janeiro, Brazil, May 2007. IEEE CS Press.

[8] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny. Pegasus: Mapping scientific workflows onto the grid. In *Grid Computing*, 2004.

[9] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, K. Blackburn, A. Lazzarini, A. Arbree, R. Cavanaugh, et al. Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*, 1(1):25–39, 2003.

[10] E. C. for Medium-Range Weather Forecasts. Seasonal forecast user guide. http://www.ecmwf.int/products/forecasts/seasonal/documentation/, 2003.

[11] Foster, I., C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation. In *7th International Workshop on Quality of Service (IWQoS), London, UK*, pages 27–36, 1999.

[12] Hoheisel, A. User Tools and Languages for Graph-based Grid Workflows. In *Workflow in Grid Systems Workshop in GGF10 at Global Grid Forum*, 2004.

[13] Krishnan, S, P. Wagstrom, and G. von Laszewski. GSFL: A Workflow Framework for Grid Services. Technical Report Preprint ANL/MCS-P980-0802, Argonne National Laboratory, Aug 2002.

[14] E. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20, 1963.

[15] Next Generation GRIDs Expert Group. Future for European Grids: GRIDs and Service Oriented Knowledge Utilities – Vision and Research Directions 2010 and Beyond. European Commission, 2006.

[16] W. M. Organization. Calculation of monthly and annual 30-year standard normals. WCDP-No. 10, WMO-TD/No. 341, Geneva: World Meteorological Organization, 1989.

[17] J. Schneider, J. Gehr, B. Linnert, and T. Röblitz. An efficient and robust protocol for reserving multiple grid resources in advance. In *3rd CoreGRID Workshop on Grid Middleware*, CoreGrid, 2008. to appear.

[18] J. Shukla. Predictability in the Midst of Chaos: A Scientific Basis for Climate Forecasting. *Science*, 282(5389):728–731, 1998.

[19] Snell, D., M. Clement, D. Jackson, and C. Gregory. The Performance Impact of Advance Reservation Meta-scheduling. In *6th Workshop on Job Scheduling Strategies for Parallel Processing, Cancun, Mexiko*, volume 1911 of *Lecture Notes in Computer Science (LNCS)*, pages 137–153. Springer, 2000.

[20] R. Souto, R. Avila, P. Navaux, M. Py, T. Diverio, H. Velho, S. Stephany, A. Preto, J. Panetta, E. Rodrigues, et al. Processing Mesoscale Climatology in a Grid Environment. *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, pages 363–370, 2007.

[21] von Laszewski, G., K. Amin, M. Hategan, and N. J. Zaluzec. GridAnt: A Client-Controllable Grid Workflow System. In *37th Hawaii International Conference on System Science*, 2004.