Meteorological Simulations in the Cloud with the ASKALON Environment

Gabriela Andreea Morar[1], Felix Schüller[2], Simon Ostermann[3],
Radu Prodan[3], and Georg Mayr[2]

[1] Babes-Bolyai University, Cluj-Napoca, Romania
gabriela.morar@econ.ubbcluj.ro
[2] Institute for Meteorology and Geophysics, University of Innsbruck
{felix.schueller,georg.mayr}@uibk.ac.at
[3] Institute of Computer Science, University of Innsbruck
{simon,radu}@dps.uibk.ac.at

Abstract. Precipitation in mountainous regions is an essential process in meteorological research for its strong impact on the hydrological cycle. To support scientists, we present the design of a meteorological application using the ASKALON environment comprising graphical workflow modeling and execution in a Cloud computing environment. We illustrate performance results that demonstrate that, although limited by Amdahl's law, our workflow can gain important speedup when executed in a virtualized Cloud environment with important operational cost reductions. Results from the meteorological research show the usefulness of our model for determining precipitation distribution in the case of two field campaigns over Norway.

1 Introduction

Scientific computing requires an ever-increasing number of resources to deliver results for growing problem sizes in a reasonable time frame. Today, Cloud computing provides an alternative by which parallel resources are no longer hosted by the researcher's computational facilities or shared as in computational Grids, but leased from large specialized data centers only when needed. To account for the heterogeneity and loosely coupled nature of resources, the scientific community adopted the workflow paradigm based on loosely coupled coordination of atomic activities as one of the most successful programming paradigms. As a consequence, numerous efforts among which the ASKALON environment [1] developed integrated environments to support the development and execution cycle of scientific workflows on dynamic Grid and Cloud environments. In this paper, we illustrate a case study of using ASKALON for porting and executing a meteorological application in a real Cloud environment. First of all, the application is specified by the user at a high-level of abstraction using a graphical UML modeling tool or an XML-based specification language. A set of advanced middleware services consisting of resource management, scheduling, and enactment

I. Caragiannis et al. (Eds.): Euro-Par 2012 Workshops, LNCS 7640, pp. 68–78, 2013.

support the transparent execution of the application on the underlying Cloud resources.

The paper is organized as follows. Section 2 gives a small introduction to the meteorological goals of our interdisciplinary research. Section 3 gives an overview of the ASKALON environment used to program and execute the meteorological application on Cloud computing resources with improved performance. Section 4 presents the workflow engineering details using the ASKALON graphical user interface. Section 5 presents performance and output research results from running our application on an academic private Cloud infrastructure. Section 6 summarizes the related work and Section 7 concludes the paper.

2 Meteorological Research

The aim of our meteorological research is to investigate and simulate precipitation in mountainous regions with a simple meteorological numerical model called linear model of orographic precipitation (LM) [2]. Applications of this model range from climatological studies to hydrological aspects. As LM is a very simple and basic model, it can be easily run in a large number of parameter studies. The model works on gridded topography and gives an analysis of precipitation resulting from a given meteorological setup. The topography can be realistic like the Alps or idealized shapes for research purposes. In the LM, a moist air mass is forced over a given topography. Lifting, which is described by simple nonhydrostatic airflow dynamics, produces cloud water. It is converted to hydrometeors and drifted with adjustable timescales.

Compared to more complex models, one big advantage of LM is the ease with which a higher horizontal resolution of the results can be achieved. Furthermore, as it is simple, a single run executes very fast compared to more complex models (i.e. in the order of minutes). Due to the fast execution time, LM allows for 1000+ instances in a short amount of time, allowing us to quickly check model sensitivity through a range of parameters or to make probabilistic forecasts by having a statistically significant number of experiments. Furthermore, no communication between the parallel instances is needed, making it an ideal application for the high latency setup of a Cloud environment, opposed to a full fledged meteorological model where a more tightly coupled parallelization (e.g. MPI) is needed. There are relatively few requirements for the parallel instances, since they only need to read small input files (a few megabytes). However the workflow activities that are collecting and extracting the results need considerably more storage I/O. The amount of data from the parallel instances needs to be condensed to be of use for a meteorological end user.

Our intent is to use the LM applications in two main areas. *(1) Research area* concentrates on investigation of the model behavior by extending the LM theory. The experiments in this area need to be very flexible, as for example not all parts of the workflow are necessarily run at each invocation. One main use is to investigate the model behavior after new theories are implemented into the model. Another option is to use it as a tool for analysis of certain meteorological phenomena or meteorological measurements. These experiments vary in the requirements to the computational infrastructure, as they vary in size, data-wise and computational-wise. The experiments of the research type are run on-demand. *(2) The operational area* aims at providing the Tyrolean avalanche service (“Tiroler Lawinenwarndienst” (LWD)) with a downscaled and probabilistic precipitation forecast, helping them with their daily issued avalanche bulletin. Taking input from a global numerical forecast model, we downscale these results to a higher horizontal resolution and test the sensitivity of the results to changes in the input parameters. The results are visualized as probability maps for the area of Tyrol. Requirements for this aspect are the availability and the reliability of the compute infrastructure. Experiments of the operational type are only run once every day for a short time, making the Cloud a more economical way to run the computations as compared to the costs of a dedicated system.

3 ASKALON

ASKALON [1] is an application development and computing environment developed at the University of Innsbruck with the goal of simplifying the development and optimization of applications that can harness the power of Grid and Cloud computing infrastructures (see Figure 1). In ASKALON, the user composes workflow applications at a high level of abstraction using a UML graphical modeling tool. Workflows are specified as a directed graph of *activity types* representing an abstract semantic description of the computation such as a Gaussian elimination algorithm, a fast Fourier transform, or an N-body simulation. The activity types are interconnected in a workflow through control flow and data flow dependencies. The abstract workflow representation is given in an XML form to the ASKALON middleware services for transparent execution onto the Grid/Cloud.

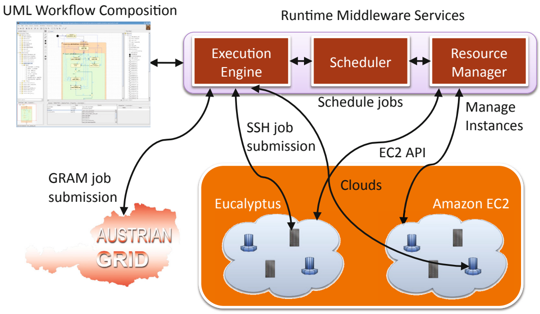


Fig. 1. Simplified ASKALON architecture extended for computational Clouds

4 RainCloud Workflow

To be able to run the original meteorological application on a distributed Grid/Cloud infrastructure, we split the monolithic simulation code in a workflow called RainCloud consisting of a set of activities described next.

4.1 Graphical Modeling

The workflow illustrated in Figure 2 receives two files as input data to be processed: `Topography.tar.gz` and `DataIN.tar.gz`. The `PrepareLM` activity has to be run by each workflow invocation and does all preprocessing of the input data. The exact nature of this depends on the flavor of the workflow invocation (see Section 4.2). Often this activity loops through ranges of model parameters and/or processes, given a topography and meteorological input data. The results of this activity are the number of `LinearModel` instances (saved in `Template_iterations.txt`) needed for future processing and the base files used for the linear model (contained in `PLM_g_out.tar.gz`). `LinearModel` is the main activity of the workflow where the meteorological model is being run as a parallel loop. `LinearModel` takes input produced by `PrepareLM` consisting of topography and meteorological control parameters and outputs a precipitation analysis. Based on the generated number of `LinearModel` instances, the workflow creates the corresponding number of parallel activities (around 1000 for a realistic simulation). To decrease the overhead for creating a large number of parallel activities and for transferring the corresponding files, we created an input parameter called `NGroup` that defines the number of linear model processes to be grouped into one `LinearModel` activity. The output of the `LinearModel` activities are `LM_g_out.tar.gz` files that can be further processed by other activities, such as `PostprocessSingle` (an optional part of the workflow) which extracts certain points from the result, condenses them to sums/means, or performs postprocessing of the results from the linear model where no comparison/combination with other model runs is needed. This activity is also used for parts of the visualization. The `LM_g_out.tar.gz` files can be further postprocessed by the `PostprocessFinalLM` activity that combines the output from different linear model runs or generates a visualization of the raw model output. The `PostprocessSingle` activity outputs `PPS_g_out.tar.gz` files that are further used as input for the `PostprocessFinalPPS` activity, which combines

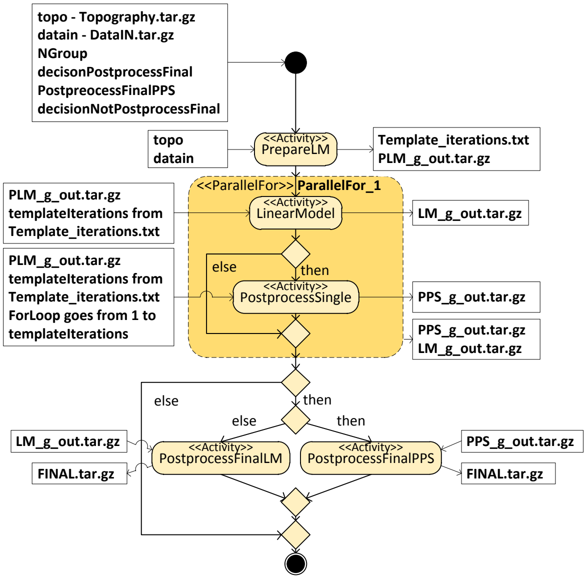


Fig. 2. Graphical representation of the workflow in ASKALON

the results from `PostprocessSingle`, performs result correction/normalization, and creates a visualization of the already postprocessed data. The output of `PostprocessFinalLM` and `PostprocessFinalPPs` is the `FINAL.tar.gz` file that contains the final results.

4.2 Workflow Flavors

We designed the workflow to be run in three "flavors". *Ideal* belongs to the operational area in which the whole setup uses idealized topography and atmospheric conditions. This workflow is mostly used for model testing or investigation of meteorological phenomena; *Semi-ideal* belongs to the research area and is designed such that either the topography is idealized or the atmospheric input is simplified, however, at least one part has real world application. For example, the topography is based on real topography data, but the atmospheric conditions are "manually" given. This flavor is used for e.g. interpretation of meteorological measurements; *Real* belongs to the research area such that both topography and atmospheric conditions are given by real-world observations or full numerical models. This flavor is used for forecasting/downscaling precipitation.

We designed the workflow such that it satisfies the requirements of the three flavors through the use of three decision nodes based of three input parameters: decisionPostprocessSingle – if true then the `PostprocessSingle` activity gets executed; decisonPostprocessFinal – if true then the `PostpreocessFinalPPS` activity gets executed, otherwise `PostprocessFinalLM`; decisionNotPostprocessFinal – if true then one of the `postprocessfinal` activities will get executed based on the evaluation of `decisonPostprocessFinal`.

5 Experiments

The goal of our experiments is twofold: (1) to investigate whether a parallel Cloud computing infrastructure is beneficial to the RainCloud workflow execution in terms of performance and costs; (2) to apply the RainCloud workflow to achieve our meteorological goals in research and operational areas.

5.1 Performance Results

The purpose of the performance experiments was to study the performance we can obtain by running the RainCloud workflow in various virtual machine configurations and on a public Cloud. We use a private Cloud that runs Eucalyptus and is based on four machines, each equipped with two Intel Xeon X5570 quad-core processors and 32 GB of main memory. Therefore our entire infrastructure provides 32 cores and 128 GB of main memory of which 112 GB are available for the Cloud. We run the workflow using different virtual machine instance types characterized by different numbers of cores (1, 2, 4, 8) and amount of memory (2, 4, 8, and 16 GB). The instance types are named from A–D with increasing numbers of cores and memory size. We continue our experiments with running the workflow on Amazon EC2 using 1 to 4 instances of the type c1.xlarge.

Throughout the rest of the paper PS1 and PS2 denote the two different problem sizes used in our experiments, with PS2 holding approximately double the computational work of PS1. In meteorological terms the different problem sizes investigate different ranges and resolutions of the targeted parameter space, in this case temperature at the ground and layer interface height of atmospheric levels.

Before performing the speedup analysis, we must emphasize that the workflow has a sequential part that limits the maximum speedup according to the Amdahl's law. Figure 3 presents a sample workflow trace snapshot obtained from the ASKALON performance analysis tool which shows that the first sequential activity takes 70 seconds and the last sequential activity 50 seconds from a total of 313 seconds of run time. This leads to the observation that for this execution, the sequential part accounts for over a third of the overall run time using 24 cores distributed to three instances.

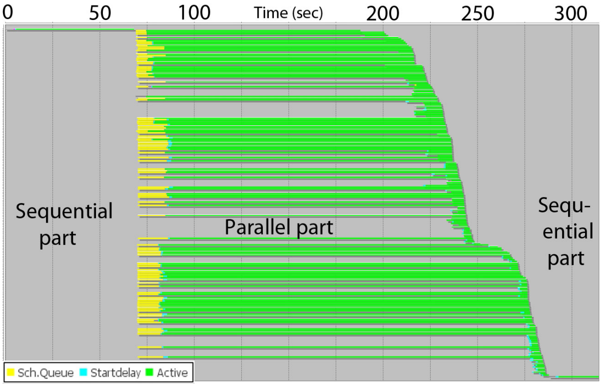
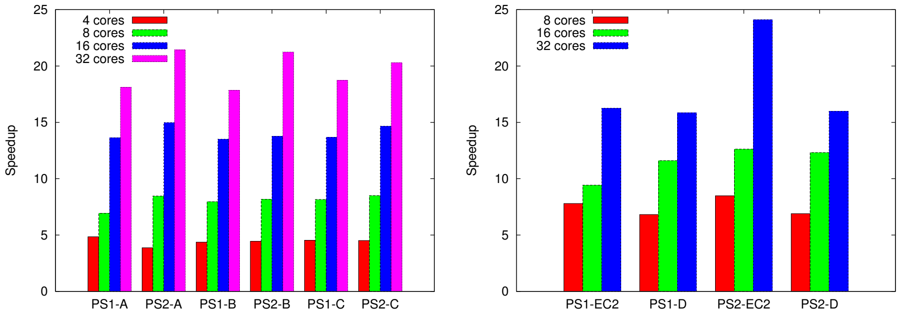


Fig. 3. Workflow execution trace snapshot using the ASKALON performance tool

Figure 4b shows the speedup of the workflow when using 1, 2 or 4 Cloud instances. We compare the performance of our private Cloud installation with configuration D (8 cores and 16 GB of memory per instance) with the results when using the Amazon EC2 c1.xlarge instances (8 cores and 7 GB of memory). The Speedup was computed using the serial execution time obtained by running the workflow on 1 core instances for the private and the public Cloud. The speedup compared to a sequential execution reaches a factor of 23 for PS2 and 32 cores of the Amazon Cloud. Our evaluation shows that the performance of the private Cloud installation is comparable with that of the public environment. This allows us to use the private Cloud for development and we can then easily switch to the public Cloud once the production runs begin.

Figure 4a depicts the speedup gained by using different Cloud instance types with a fixed ratio of 2GB memory per core, where the sequential time was computing by running the workflow on a one-core instance using the corresponding amount of memory. The experiments were run for both problem sizes.

As illustrated, the results show that the application is scalable, reaching a maximum speedup of 21.4 for PS2 with instance type A and 32 cores. In addition the results indicate that the application scales better for higher problem sizes, which was the case for almost all our experiment runs. The maximum speedup



(a) Speedup for different sizes of private Cloud instances.

(b) Private Cloud versus Amazon EC2.

Fig. 4. Speedup analysis. PS# denotes the problem size and A–D and EC2 the instance type.

increase of PS2 over PS1 is 22%; the average increase is 4.3%. As Figure 4a illustrates, varying the instance types only has a small effect on the speedup, with a maximum speedup increase of 20% for instance type C over D for PS1 using 8 cores, and 9% for instance type A over B for PS2 using 16 cores.

In general, the execution of one workflow with the experimental input data would cost approximately $2.72 if executed on Amazon EC2 using 16 `c1.medium` instances ($0.17/hour). This result applies to all presented workflow instances, as their execution time is lower than the one-hour payment granularity of EC2. For a yearly cost of $992.8 this workflow can be run once every day, which is only a fraction of the amount the purchase of a comparable, dedicated system would cost.

5.2 Meteorological Results

In this section we show the meteorological results obtained by conducting two experimental runs using the RainCloud workflow: (1) an explorative run for optimizing a future experiment setup, and (2) a simulation of precipitation in the region of the Kongsvegen glacier, Svalbard, Norway.

Explorative Study. Extensive precipitation measurements taken during the 2006 STOPEX2 field campaign over Stord on the west coast of Norway provide ground truth for numerical simulation studies. In order to determine how large the spatial domain of these simulations has to minimally be in order to successfully simulate the flow and precipitation amounts, an explorative study investigates how strongly the precipitation of Stord is influenced by the downstream main "ridge" of Norway. For this experiment, we used a pseudo-three-dimensional cross section through the center of Stord using the ideal RainCloud workflow flavor (see Figure 5(a)). We run the model with two atmospheric layers where winds differ. We changed the length of the topography within each of these experiments

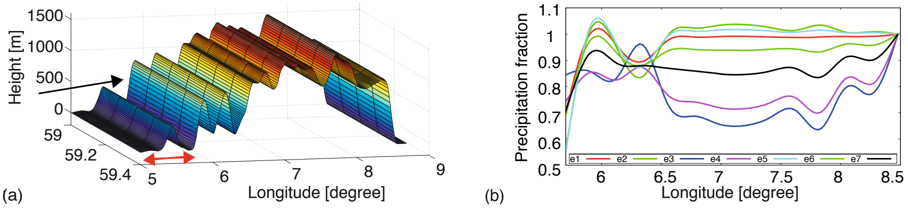


Fig. 5. (a) Topography [meters] used for the explorative study. The black arrow indicates wind direction, the red arrow the target area of Stord. Latitude is degrees north, longitude degrees east. (b) Precipitation sum over Stord (5.5-5.7 degrees) from varying topography lengths as fraction of the reference precipitation. The experiments are (upper level wind speed/lower level wind speed in m/s): e1 (15/15), e2 (5/15), e3 (45/15), e4 (45/45), e5 (5/5), e6 (25/25), e7 (35/35).

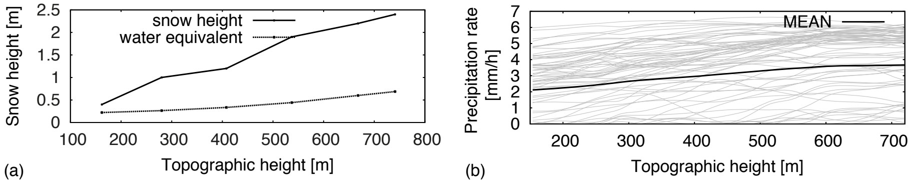


Fig. 6. (a) Measured snow height and water equivalent along topographic height of the Kongsvegen glacier during winter 2010/2011. (b) Precipitation rates from LM simulations with varying wind directions and speeds in respect to topographic height.

from only having Stord (5.5-5.7 deg E) up to the full reference length (5.5-8 deg E). Both variations lead to 690 instances of the activity `LinearModel` with a small input size per instance. Activity `PostProcessSingle` extracts a precipitation sum over the area of Stord between 5.5 and 5.7 deg E for each model run, which is then compared with the precipitation amount from the reference run in `PostprocessFinalPPS`. A subset of simulations is shown in Figure 5(b). Higher wind speeds lead to an increasing importance of the downstream mountains. For small wind velocities, we could cut the topography at around 6.8 deg longitude, accepting a small error of about 2 percent. When higher wind speeds occur, the complete Norwegian ridge needs to be included in activity *LinearModel*.

Snow Accumulation over a Norwegian Glacier. The research workflow setup was further used for an investigation of precipitation/snow accumulation on the Kongsvegen glacier on Svalbard, Norway (SvalClac project). Snow heights and water equivalents were derived from snow density measurements within snow pits on the central flow line of the Kongsvegen glacier. These measurements show a nearly linear gradient with topographic height. The question to be answered with our workflow simulations was whether this gradient is explainable by orographic precipitation effects. A topography with 150m resolution and meteorological values from a typical precipitation event (5th Nov. 2010) are used. In a first set of experiments we varied wind speed from 0m/s to 40m/s and direction from 0 and 360 degrees, leading to 900 instances of `LinearModel`. One instance has a relatively big input size of 20Mb due to the high-resolution topography. Figure 6 (b) shows the simulated precipitation along the glacier extracted by `PostProcessSingle` (no `PostprocessFinal` is run). The snow measurements are an "integral" over the whole winter including various wind directions and speed, therefore we apply a mean over all model results including those without precipitation (black line). It shows that the model exhibits a similar behavior as the measurements, indicating that the gradient is explainable by orographic precipitation effects. To eliminate other possible causes further investigation is necessary.

6 Related Work

The scientific community shows growing interest in Cloud computing. The FutureGrid [3] project provides a scientific Cloud infrastructure on demand for scientists to experiment with this environment without the cost that commercial Cloud providers charge for their usage. The infrastructure is only available for research purposes and production runs are strictly prohibited. The work in [4] shows that Cloud computing can be used for scientific workflows, with the Montage workflow. This proof of concept motivated the community to start exploring this resource type as a Grid alternative. Other approaches [5] use Cloud resources in a different way, e.g. by extending Clusters with additional resources from a Cloud during peak usage, increasing the throughput of the system. A similar extension of the Torque job manager to add Cloud resources to clusters is presented in [6]. Our approach does not rely on any existing clusters and tries to optimize the workflow execution on a Cloud-only environment. The research presented in [7] shows a workflow engine developed for Cloud computing. We have the advantage of a mature workflow engine, which evolved in the Grid area and was extended for a hybrid usage of both technologies. In this paper we use it with Clouds only to show that it is well optimized for this resource class as well. The Megha workflow management system [8] is designed to execute scientific applications, designed in the form of workflows described in xWFL, on Cloud and Grid resources. It is created in the format of a portal. The authors prove that by tuning the applications to run on Amazon EC2 cloud resources time consumption can be significantly reduced. Aneka [9] is a platform and a framework for developing distributed applications on the Cloud. It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or data centers on demand. A parallel among the performance of execution of scientific workflows on commercial cloud resources (Amazon EC2) and HPC systems (NCSA's cluster) is presented in [10]. Three different workflows are tested: Montage, Broadban and Epigenom. The performance is similar, although a bit lower for the Cloud, due to less powerful EC2 resources.

7 Conclusions

We illustrated a case study of using ASKALON for porting and executing a real-world meteorological application called RainCloud in an academic private Cloud environment. The application designed as a workflow implements a highly simplified linear model of orographic precipitation to answer meteorological research questions in connection with measurements from two field campaigns in mountainous areas. We presented the application design performed at a high-level of abstraction using a graphical modeling tool. A set of advanced middleware services comprising resource management, scheduling, and enactment support the transparent execution of the application on the underlying Cloud resources. We present performance results that demonstrate that, although limited by Amdahl's law, our workflow application can gain important speedup when executed in a virtualized Cloud environment with significant cost reductions if operated in a production environment. Performance results achieved by using a private Cloud and those attained with Amazon EC2 instances show that we can use the private Cloud for development purposes and then are able to switch to a public Cloud when production runs will begin. Results from the meteorological research show that our simplified model is a useful tool for determining possible causes for precipitation distribution in the case of two field campaigns over Norway.

Acknowledgment. The research was funded by the Standortagentur Tirol: RainCloud and data from the European Centre for Medium-Range Weather Forecasts was used for the meteorological application. G.A. Morar thanks the: Investing in people! Ph.D. scholarship, contract nr. POSDRU/88/1.5/S/60185.

References

1. Fahringer, T., Prodan, R., et al.: Askalon: A development and grid computing environment for scientific workflows. In: Scientific Workflows for Grids. Workflows for e-Science. Springer (2007)
2. Barstad, I., Schüller, F.: An Extension of Smith's Linear Theory of Orographic Precipitation: Introduction of Vertical Layers. Journal of the Atmospheric Sciences 68(11), 2695–2709 (2011)
3. Riteau, P., Tsugawa, M., Matsunaga, A., Fortes, J., Keahey, K.: Large-scale cloud computing research: Sky computing on futuregrid and grid'5000. ERCIM News (2010)
4. Hoffa, C., Mehta, G., Freeman, T., Deelman, E., Keahey, K., Berriman, G.B., Good, J.: On the use of cloud computing for scientific workflows. In: eScience, pp. 640–645. IEEE Computer Society (2008)
5. Assuncao, M., Costanzo, A., Buyya, R.: Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In: Kranzlmüller, D., Bode, A., Hegering, H.G., Casanova, H., Gerndt, M. (eds.) 11th IEEE International Conference on High Performance Computing and Communications, HPCC 2009. ACM (2009)
6. Marshall, P., Keahey, K., Freeman, T.: Elastic site: Using clouds to elastically extend site resources. In: CCGRID, pp. 43–52. IEEE (2010)

7. Franz, D., Tao, J., Marten, H., Streit, A.: A workflow engine for computing clouds. In: The Second International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2011, Rome, Italy, p. 6 (2011)
8. Pandey, S., Karunamoorthy, D., Gupta, K.K., Buyya, R.: Megha Workflow Management System for Application Workflows. IEEE Science & Engineering Graduate Research Expo (2009)
9. Vecchiola, C., Chu, X., Buyya, R.: Aneka: A Software Platform for .NET-based Cloud Computing. IOS Press (2010)
10. Juve, G., Deelman, E., Vahi, K., Mehta, G., Berriman, B., Berman, B.P., Maechling, P.: Scientific workflow applications on amazon ec2 (arXiv:1005.2718) (May 2010)