Fostering Energy-Awareness in Simulations Behind Scientific Workflow Management Systems

Gabor Kecskemeti*†, Simon Ostermann†, Radu Prodan†
*Institute for Computer Science and Control of the Hungarian Academy of Sciences, Hungary
†Institute of Computer Science, University of Innsbruck, Austria
Email: kecskemeti.gabor@sztaki.mta.hu, {simon,radu}@dps.uibk.ac.at

***Abstract*—Scientific workflow management systems face a new challenge in the era of cloud computing. The past availability of rich information regarding the state of the used infrastructures is gone. Thus, organising virtual infrastructures so that they not only support the workflow being executed, but also optimise for several service level objectives (e.g., maximum energy consumption limit, cost, reliability, availability) become dependent on good infrastructure modelling and prediction techniques. While simulators have been successfully used in the past to aid research on such workflow management systems, the currently available cloud related simulation toolkits suffer form several issues (e.g., scalability, narrow scope) that hinder their applicability. To address this need, this paper introduces techniques for unifying two existing simulation toolkits by first analysing the problems with the current simulators, and then by illustrating the problems faced by workflow systems through the example of the ASKALON environment. Finally, we show how the unification of the selected simulators improve on the the discussed problems.**

***Keywords*—Scientific computing, distributed computing, simulation, workflow**

I. INTRODUCTION

Scientific workflows [1] enable constructing and executing large scale distributed applications based on well understood basic building blocks, designed for scientists with less expertise in organising and enacting a complex application. The burden of organisation and enactment lies on the underlying workflow management systems, that must not only ensure the proper and timely execution of the users' complex applications, but should also optimise their distribution and schedule on the available infrastructures. With the advent of cloud computing [2], workflow management systems must not only cope with the available infrastructures, but must be able to decide when and how to improve user experience with the inclusion of leased virtual infrastructures.

Although the building blocks of these scientific workflow applications could have executions in the range of months, the way they are enacted by the workflow systems could have significant effects both on their runtime as well as on the underlying infrastructures [3]. In the past, several workflow systems used simulators [4], [5] to evaluate the possible effects of particular enactment scenarios on workflows and infrastructures. Simulations are important tools to speed up research evaluations that otherwise would need too much time in reality. The increase in speed is normally reached by simplifying the model of the system to be simulated trying to stay as close to reality as possible. Simulations in some extreme cases are increasing evaluation speed to such levels that they allow close to real-time evaluation of multiple situations. Unfortunately, past workflow management techniques, which were incorporating simulators in their decision making process, hardly considered the highly volatile and dynamic nature of cloud systems.

Although several cloud simulators exist today [6], [7], [8], they cannot support the requirements of current workflow management systems. They are frequently focused on the user side, therefore mostly considering clouds as a black box. Unfortunately, this behaviour does not allow the analysis of the infrastructure level effects of the various decisions made by the workflow management systems. Even in such cases, when a simulator offers insights on how clouds internally operate, they are focused on specific areas (e.g. providing accurate CPU or network sharing, energy modelling) and are often less developed in others; therefore they restrict the use cases in which these simulators would be useful for the complex decision making process [9] in cloud aware workflow management systems.

Through the example of a well researched scientific workflow management system (namely ASKALON [10]), we analyse in this paper the possible improvements one could gain by integrating a user-side simulator (called GroudSim) with a an internal infrastructure focused simulator (called DISSECT-CF). Using this approach, we can not only fulfil the demands of current research directions but also allow the widening of research applied in scientific workflow management systems. Thus, this paper has two distinct contributions: (i) the integration of two complete simulators in a way that keeps their features while minimising the overhead caused by their joint operation, (ii) the analysis of new research directions the merged simulators could offer to the community researchers responsible for scientific workflow management systems.

We have chosen the ASKALON system because it has already been integrated with the GroudSim [7] simulator to support in its workflow enactment related decisions. For the role of the second simulator, we have selected a versatile simulation framework capable to simulate the internals of the cloud infrastructures allowing the evaluation of energy consumption, network behaviour and the effects of cross virtual machine CPU sharing. The selected simulation framework is called DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations (DISSECT-CF) [11]. Although we have evaluated the integration on these specific systems, our carefully executed extensions show that the introduced techniques would be applicable to similar workflow systems too [12]. Our extensions show that an existing workflow system could already benefit from such integrated simulations with minimal or no changes to its workflow management techniques.

The rest of the paper is organised as follows: In section II we show the currently existing systems and describe the advantages our approach has. The background information needed to understand the existing system is given in section III followed by details about the integration in section IV. We summarise the new possibilities achieved by this extension in section V and conclude the paper with a short outlook to upcoming work in section VI.

II. RELATED WORK

Scientific applications are complex systems consisting of different programs that often need days or weeks to be executed. Users of such applications apply the workflow paradigm or other techniques to increase parallelism and achieve faster execution times. To research the impact of different schedules or optimisations, simulators are often employed to reduce the time between implementation of features and their verification.

The surveys in [4], [5] give a good overview of simulators, some covering the field of Cloud computing. The status of GroudSim in the [4] survey shows important missing features, while some features must have been overlooked by the authors (a cost model exists in GroudSim since its initial version, and was extended over the years to support all commercially available billing models). Other crucial features provided by DISSECT-CF on the internals of infrastructure clouds (e.g. energy models, more complex networking) are introduced in this publication and have been added into GroudSim.

GridSim [13] and its extension CloudSim [6] are well known simulation environments for task executions on Grid and Cloud platforms. As our previous work showed, the scalability and flexibility is their biggest problem [7].

iCanCloud [8] is a new contribution to the area of Cloud simulators specialised on Amazon EC2 resources using a configuration GUI. Because of its user orientation, iCanCloud lacks crucial functionality needed for Green IT research [14] such as power consumption, and has no workflow support.

WorkflowSim [15] is an open source workflow simulator that extends CloudSim by providing new constructs for simple management and simulation of workflows. It models workflows as a DAG and provides out of the box implementation for several popular workflow schedulers (e.g., HEFT, Min-Min) and task clustering algorithms. Its main disadvantage (alongside its limitation to the DAG model lacking loops) is that it misses a connection to a real-life workflow management system such as ASKALON.

SimGrid [16] has been developed over the past years as a versatile, accurate and scalable simulator. Compared to our solution, it lacks support for dynamic workflow applications, as it only supports static DAGs, and does not offer important features like real life and simulated executions within the same environment. Researching new methods and ideas needs therefore twice effort required in ASKALON: first the validation must be performed in SimGrid, and afterwards the new code needs to be rewritten for the real execution environment.

Compared to other existing simulators, two features make the combination of the ASKALON-GroudSim system with DISSECT-CF unique: (i) the possibility to simulate and execute workflow applications directly within the same environment, and (ii) the integration of a unified power utilisation and resource sharing model regarding the simulated data centre components.

III. BACKGROUND

Simulation is a known useful practice when trying to solve complex problems like scheduling, resource management or workflow executions. There are multiple tools available for this purpose, as mentioned in section II, but they either lack functionality or are not user friendly. Especially the high interest in power-aware methods is not satisfactory with the current available simulators in the scope of workflow executions on cloud resources. To overcome this drawback, we developed a simulator specialised on Infrastructure as a Service (IaaS) clouds with focus on power consumption and scalability of the simulation. We integrated this simulator into an existing framework for workflow development, execution and simulation called ASKALON.

A. ASKALON

ASKALON, an existing middleware researched at the University of Innsbruck, provides an integrated environment to support the development, simulation and execution of scientific workflows on dynamic Grid and Cloud infrastructures [10]. Figure 1 shows the design of the ASKALON system with focus on the integrated simulator, explained in detail in subsection III-B. Workflows can be graphically programmed in an abstract and user-friendly fashion and submitted for execution to the ASKALON services that shields the users from the low-level cloud infrastructure technology details. A command line client also exists to allow script based batch execution of workflows in an XML language representation. Execution information is stored in a database allowing online and post-mortem analysis of workflow runs using a graphical tool or SQL-queries. The three main components that handle the execution of workflows are explained in the following.

Execution Engine. Submission of jobs and transfer of data to the compute resources is done with a suitable protocol, e.g. ssh for Cloud resources or GRAM in a Globus/Grid environment. For simulated workflow executions, we developed a new provider for the Globus COG-kit that allows the use of the existing abstraction model to interact with the integrated simulator. There is also a scheduling module included in this part that allows its main logic to be easily replaced with different new algorithms.

GridARM/GLARE. The resource manager has the task to provision the correct amount of resources at the correct moment to allow the execution engine run the workflow as decided by the scheduler. For this purpose, it communicates with different Cloud providers, or in the simulation case with GroudSim, to provision cloud instances using predefined images for the required applications.

GAB. The GroudSim-ASKALON-Bridge is responsible to distinguish between the real processes run by ASKALON and the simulated environment operated by GroudSim. As ASKALON is used for executions on real hardware, this module was needed to allow the integration of the simulator in a transparent fashion to the other components

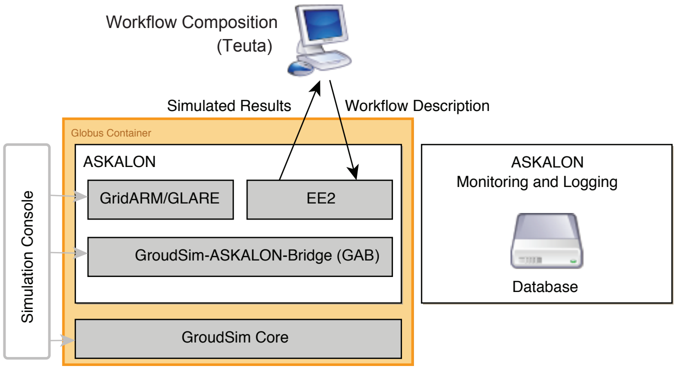

Fig. 1. The architecture of ASKALON

that are not simulated such as the scheduler, resource manager and job submission. This module ensures that the simulation time is only advanced when no more new events from ASKALON are generated.

B. GroudSim

GroudSim is an event-based simulation toolkit for scientific applications running on combined Grid and Cloud infrastructures developed in Java. GroudSim uses a discrete-event simulation toolkit that consists of a future event list and a time advance algorithm that offers improved performance and scalability compared to other process-based approaches used in related work [17]. The simulator can be used in a stand-alone fashion or integrated in the ASKALON environment. The later option allows seamless development, debugging, simulation and execution processes using the same ASKALON interface offered to the end-users.

Figure 2 shows the most important components of GroudSim, which collaborate internally and act as interfaces to communicate with the GAB. The two central parts of the simulation framework are: (i) the *event system* storing information about the type and time of the events, and (ii) the *simulation engine* responsible triggering the events at well-defined moments in time. Events can simulate job executions, file transfers, availability of resources (including failures), and background load. The other GroudSim core components are:

Resource module that manages the simulated resources and communicates them to GridARM/GLARE;

Synchronisation module which allows synchronisation of the simulation time and the time used by the execution engine. When the engine is generating new tasks and submits them to the simulator, the simulator must wait until all current tasks are submitted before the simulation time can be advanced;

Background loader adds additional load to the resources upon the request of the user or the GAB. The load can be achieved by using traces from the Grid workload archive or by using synthetic job distribution functions;

Failure generator which handles the failure rates for jobs, file transfers and resources following stochastic distributions;

Stochastic framework that offers different stochastic distribution functions, which can be used for calculating queuing times, submission times, execution times, failure rates or background loads.

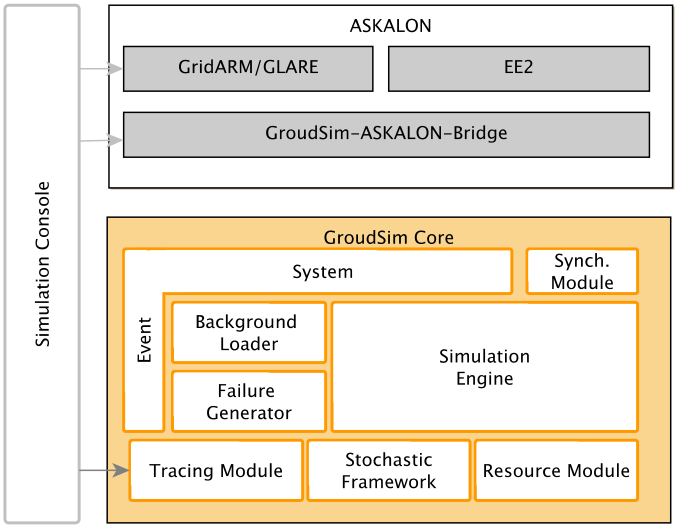

Fig. 2. The architecture of GroudSim

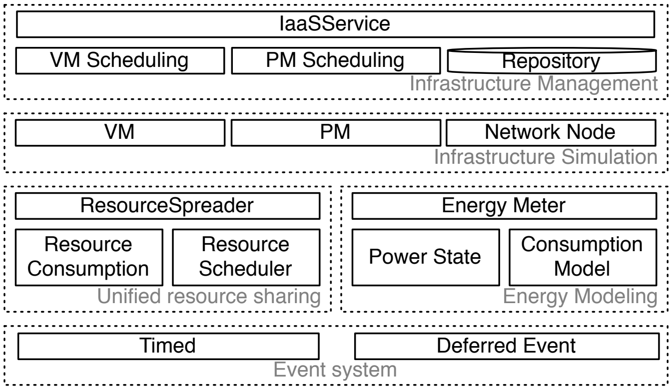

Fig. 3. The architecture of DISSECT-CF

Tracing module which is used to store the simulated execution events to a file for analysis or debugging.

In addition to these service components, the GroudSim enabled execution and simulation service provides a GUI that allows easy setup of the Grid and Cloud resources by showing statistical charts of the simulated tasks and file transfers. All this information can also be collected in the performance database or setup via configuration files, if preferred.

C. DISSECT-CF

As we plan to support future research regarding ASKALON workflow enactors, we aimed at increasing the capabilities of GroudSim with the least effort. Unfortunately, GroudSim's lack of internal IaaS behavioural knowledge reduces the number of future use cases that could be supported in the ASKALON-GroudSim system. Therefore, we have analysed several simulators that could act as the foundation for GroudSim and offer insights about the internals of IaaSs. Since GroudSim's focus was primarily on speed and efficiency we looked for a simulator that is also a good performer while it has similar internal concepts of time, events and infrastructure components. Finally, we have chosen DISSECT-CF for this purpose.

DISSECT-CF [11] is a compact, highly customisable open source cloud simulator with special focus on the internal organisation and behaviour of IaaS systems. Figure 3 presents the architecture of the currently available[1] 0.9.4 version. The figure groups the major components with dashed lines into subsystems. Each subsystem is implemented as independently from the others as possible. There are five major subsystems each responsible for a particular aspect of internal IaaS functionality: (i) event system – for a unified time reference; (ii) unified resource sharing – to resolve low level resource bottleneck situations; (iii) energy modelling – for the analysis of energy usage patterns of individual resources (e.g., network links, CPUs) or their aggregations; (iv) infrastructure simulation – to model physical and virtual machines as well as networked entities; and finally (v) infrastructure management – to provide a real life cloud like API and encapsulate cloud level scheduling.

After the simulators are integrated, new ASKALON workflow enactors are expected to improve because they can utilise more information than just the previously available job runtimes and virtual machine execution prices. Thanks to DISSECT-CF, the new enactors will be capable to use virtual machine instantiation timings, job/VM or even workflow level energy consumption details and a more precise network and CPU process model. On the other hand, if used through GroudSim, DISSECT-CF will immediately gain cloud pricing capabilities and the possibility to involve hybrid workloads utilising both clouds and grids in a single simulation. The following section will detail how the integration of the two simulators enable these new functionalities.

IV. INTEGRATION

Throughout the integration, we have aimed at maintaining API compatibility of GroudSim, thus ensuring that past work on GAB does not need to be repeated. We have investigated the APIs of both GroudSim and DISSECT-CF and we have analysed the bridging functionalities needed to cross simulator boundaries. According to our analysis, there are three major areas where the simulators have significantly differing but relevant APIs for our goals to enable more sophisticated simulation based workflow enactment. These three areas are the following: (i) the event systems have different event types, event firing mechanisms, clock maintenance techniques; (ii) cloud representations have conceptual disagreements on data centre organisation, VM and job management mechanisms; and finally (iii) network construction, utilisation, sharing and organisation. The rest of this section discusses how the gaps amongst these areas were closed allowing us a seamless transition from GroudSim level simulation to the abstraction used in DISSECT-CF.

A. Event systems

First of all, we have chosen to make GroudSim as the master simulator. As a result, there should not be events in DISSECT-CF unless there was a preceding GroudSim event that caused a series of DISSECT-CF ones. Second, if some activity happens in DISSECT-CF that has an equivalent in GroudSim, then it must be ensured, that DISSECT-CF level events are never sent directly to the user of GroudSim. Instead, they must set off an equivalent GroudSim event (see Figure 4's GroudSim event generation activity). This technique ensures, that simulations utilising GroudSim features never need to be aware of the internals of the DISSECT-CF based activities, while the technique also reduces the number of events that must go through GroudSim and thus increases the performance of the integrated simulators.

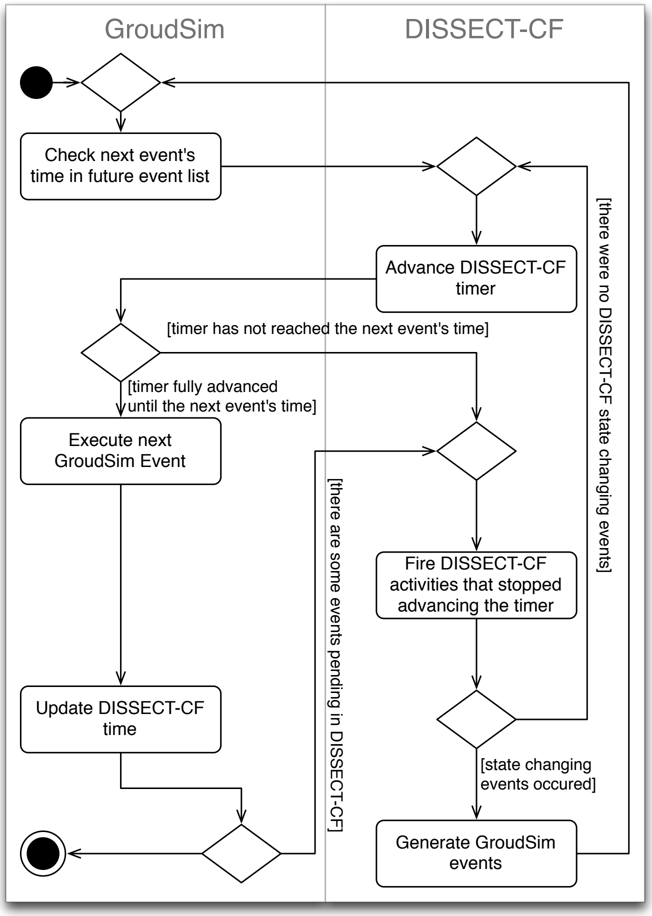

Fig. 4. Interaction between GroudSim and DISSECT-CF within the main event processing loop of GroudSim's simulation engine. *Remark: the diagram only captures the processing of a single GroudSim event within the loop*

Now that we have seen how events could occur cross simulation boundaries, let us focus our attention on the way the timing of these events are also managed in both simulators simultaneously. To keep the two simulators in synch, we have chosen to extend the simulation engine of GroudSim. This extension alters the simulators future event list processing and inside its event loop it always ensures that at any given time instance neither DISSECT-CF nor GroudSim has events, which should have happened already according to the maintained time in the other simulator. The new extension is depicted in Figure 4. The time of GroudSim is kept in sync with DISSECT-CF by ensuring that only GroudSim's simulation engine controls the time of the underlying DISSECT-CF simulation (see the time advancement and update activities in the Figure). The extension also handles situations when events in one of the simulators cause events in the other one (e.g., see the last

---

[1] https://github.com/kecskemeti/dissect-cf

conditional activity on the side of GroudSim in the Figure). This is especially important as DISSECT-CF has two kinds of events: time- and state dependent ones. Time dependent events are placed in the event queue of the *Timed* class, but state dependent events are fired by the entities that have had their states observed. In GroudSim, these two kinds of events are linked with the technique of event references and during their creation every event has its occurrence time predetermined. However, to reduce the synchronisation overhead, we choose not to create event references in sync with GroudSim (as the occurrence times are not yet available for state dependent events). Instead, when a state dependent event occurs, we request GroudSim to insert a new event into its queue for immediate execution (see the end of the DISSECT-CF activities in Figure 4).

B. *Cloud representation*

Originally, infrastructure clouds were conceptually differently simulated in the two simulators. While GroudSim focused mostly on the blackbox cloud model, DISSECT-CF offered insights on the internals of IaaSs. The blackbox model allowed GroudSim to abstract away such activities like virtual machine creation details, virtual machine placement and physical machine state scheduling. This model ensured the performant evaluation of cloud related workloads. Unfortunately, the blackbox model cannot be applied successfully to cloud infrastructures with limited resource capabilities such as private or academic clouds because the abstracted activities could make significant differences to the outcomes of virtual machine operations. Thus, we choose to keep the APIs of GroudSim, but dropped the blackbox model and ensured that DISSECT-CF simulates the previously abstracted functionalities. Although, this addition introduces some performance penalties, we have chosen DISSECT-CF because it has been shown to be a better performer than other simulators with similar features.

1) Cloud infrastructure management: Because of GroudSim's blackbox approach, clouds are defined by two properties: the number of cpu cores and the set of virtual machine instance types one can create on top of the cloud. On the other hand, in DISSECT-CF, one can define the kinds and the amounts of physical machines that constitute the cloud, and it is also possible to set energy consumption properties, custom virtual machine and physical machine schedulers. The integrated version introduces more flexibility to GroudSim's cloud representation on the following two approaches: (i) limited customisability restricted to the number and kind of physical machines; and (ii) extended customisability that enables better energy awareness through customisable consumption properties and physical machine schedulers. Unfortunately, even with the extended approach the customisation of virtual machine schedulers is not entirely possible because GroudSim expects clouds to reject virtual machine instance requests that cannot be served in the current state of the simulated IaaS. This behaviour is similar to what one can expect from commercial cloud systems and several academic cloud wares (like OpenNebula) currently. Thus, it is supporting the research on such IaaSs that are available today. The evaluation of workflows in future IaaS constructs is not supported without conceptual changes in GroudSim's cloud representation.

Next, we are going to detail the approach of limited customisability. In this case, the user of GroudSim is not expected to know that at the background there is another simulator for the internals of cloud infrastructures. In such case, we expect that users first define what kind of instances they will need from a particular cloud. Our approach then determines the maximum number of CPUs, the top performance (in terms of MIPS/core), and the biggest amount of memory needed by any of the user defined instances. These maximums are used for the definition of the template physical machine which will be the foundation of the DISSECT-CF cloud infrastructure. In DISSECT-CF, we will create as many of these kind of physical machines as many can match the amount of CPU cores asked by the user for the particular cloud during its construction. The created physical machines will all be connected together via a cloud level network and the internal DISSECT-CF cloud representation will also simulate a single repository to store a single kind of virtual appliance from which all the virtual machines can be derived. The physical machines will be controlled by a physical machine scheduler that keeps them always on. As it can be seen, this infrastructure is rather limited and as a result it also seriously limits the possible evaluation scenarios the integrated simulators can support.

To remove some of these limitations, the simulator also allows the loading of the internal cloud's properties via a file. In this file, users can define the topology of the physical machines, also they can provide custom power profiles to them and finally they can change the physical machine schedulers as well. These alterations enable network and energy aware workflow enactment, but demand user knowledge about the creation of the cloud description file that DISSECT-CF can process with its *CloudLoader*. Fortunately, the cloud description file allows us to keep the GroudSim APIs unchanged and to alter IaaS behaviour from one simulation run to another by just changing these descriptors.

2) Binding between the two VM representations: DISSECT-CF allows flexible and continuous resource constraint control during its virtual machine instance creation mechanism (i.e., users can ask for arbitrary cpu, memory and processing capabilities for their future virtual machines). This is similar to the behaviour of several academic cloud wares. Unfortunately, the instance type system of GroudSim significantly limits the possible kinds of virtual machine instances one can create similarly to how Amazon EC2 limits their users. To keep the Cloud concept of GroudSim, in the integrated simulation, we have limited the continuous resource constraint space of DISSECT-CF to the instance types from GroudSim.

To handle grids and clouds uniformly, GroudSim considers a single virtual machine as a *CloudSite*. Therefore, such resources are scheduled by OS level schedulers instead of local resource management systems applied in grid systems. In GroudSim, CloudSites are requested with an instance type. As depicted by Figure 5, this request is then forwarded to DISSECT-CF. Where the simulation schedules the VM to the most suitable physical machine. If the current physical machines in the cloud are too loaded and cannot serve the requested instance type, the VM scheduler will mark the requested virtual machine as non-servable, allowing DISSECT-CF to fail the CloudSite acquisition process in GroudSim. If the VM request can be allocated to a physical machine, the VM is instantiated on it (by simulating the transfer of its

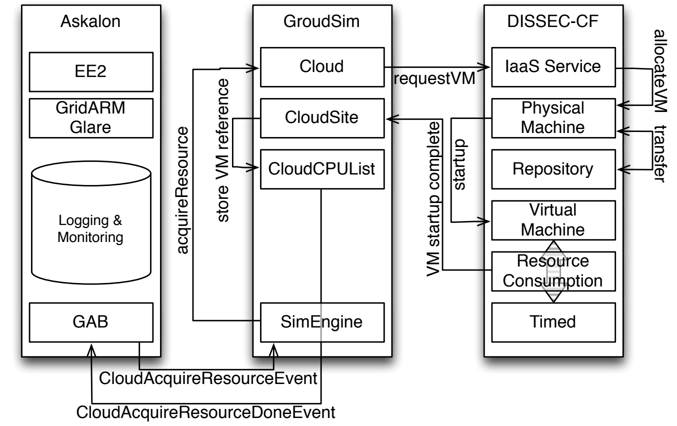

Fig. 5. Extended VM instantiation procedure

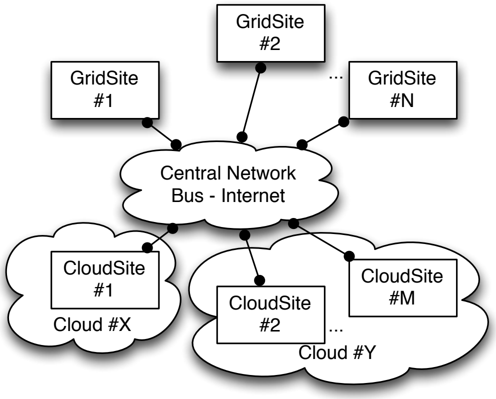

Fig. 6. Original GroudSim network topology

virtual appliance to the necessary storage element and then by simulating its startup procedures). Finally, the GroudSim user is notified about the creation of the new VM.

It must be noted, that in GroudSim, one cannot provide any relevant information to differentiate the planned function of newly created virtual machines. As a result, currently GroudSim always instantiates DISSECT-CF virtual machines with the same virtual appliance. As appliance size is a significant factor in virtual machine creation time, this loss of differentiation between virtual appliances reduces the variance of virtual machine creation times significantly. Therefore, even in DISSEC-CF enhanced GroudSim simulations, the variance of a particular appliance's transfer can be affected only by network activities like transfers between virtual machines or significant virtual machine creation bursts. Later on, we will further extend GAB so it will be able to forward the expected functionality of a future VM by sending the properties of the applications planned to be run on the VM under creation.

3) Job scheduling: Since GroudSim did no mapping between physical and virtual machines, there was no chance to observe several phenomena that occurs in under-provisioned clouds. CloudSites processed jobs independently from other CloudSites in the particular cloud infrastructure, despite they could share resources in the background. This sharing reduces the accuracy of GroudSim in scenarios involving heavy cloud usage.

Jobs in GroudSim are also restricted to use a single cpu core. In grid sites this restriction is further extended so one CPU is not allowed to have multiple jobs. But GroudSim removes this restriction for clouds. As a result, GroudSim allows the simulation of simple VM level resource bottlenecks. Unfortunately, this bottleneck situation is less frequent in clouds, especially with job models when one cannot reduce or suspend the processing of a job if needed (e.g., because of changing application characteristics or job migration across VMs).

The above mentioned issues hinder the evaluation of advanced scheduling and workflow enactment techniques applied in ASKALON. Thus, during the integration, we have aimed at removing these limitations. With its unified resource sharing mechanism, DISSECT-CF offers a widely applicable resource scheduling technique that can efficiently and more accurately manage resource bottleneck situations.

After the integration, a GroudSim simulates a job in two phases. First, GroudSim manages the job's lifecycle until it should be running on a CloudSite. In that case, the selected CloudSite injects a new CPU level resource consumption into the virtual machine representing the site in DISSECT-CF. Then comes the second phase of job execution: DISSECT-CF applies its resource sharing technique that automatically considers both the physical machine's load, which hosts the VM, and also the currently processed jobs in the VM. If a simulation set up a cloud infrastructure with a virtual machine scheduler that allows under-provisioned virtual machines, then the jobs will experience performance drops automatically. Also, the integrated simulators allow jobs to have limited performance for some periods of time and cancellation free job migration across other DISSECT-CF simulated virtual machines.

C. Networking

GroudSim have offered customisable network links among both grid- and cloud sites. These links are connected to GroudSim's central bus representing the Internet (see Figure 6). *FileTransfers* between two sites then passed through two network links and the Internet. When bandwidth was utilised by multiple file transfers, the share of each transfer was estimated based on the network link with the smallest bandwidth. This estimate however often lead to unused network capacities and highly inaccurate network bandwidth utilisation compared to real life or packet level simulator results.

Although, DISSECT-CF still offers a simplified network model, it can model GroudSim's central bus topology without any modifications. Also, thanks to its unified resource sharing model, it can immediately offer a solution to network resource bottleneck resolution. Clouds loaded with extended customisability can even limit their overall connections to GroudSim's central bus as a whole. As a result, DISSECT-CF extended simulations can organise GroudSim's clouds and grids into a hierarchical network (see Figure 7) where new workflow enactment techniques could investigate the effects of alternative cloud deployment layouts on network transfers, latencies and virtual machine instantiation times.

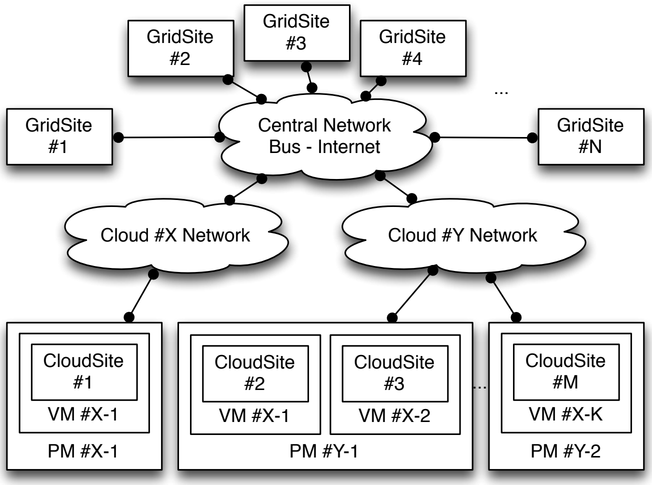

Fig. 7. Integrated GroudSim-DISSECT-CF networking

V. IMPROVED ASKALON BEHAVIOUR

A. *Scalability experiments*

After the integration of DISSECT-CF into GroudSim, we first aimed at determining the performance and scalability penalties introduced because of the additional features. We expected the performance drop because our time synchronisation and DISSECT-CF's more detailed infrastructure simulation techniques. In order to evaluate the properties of the integration, we have chosen a typical simulation scenario: simulating realistic background loads with the help of GroudSim's background loader and the Grid Workload Archives (GWA). We have chosen to use the background loader as it is capable to do larger scale and more realistic simulations in contrast to the use of a single or repeated workflow execution otherwise possible through ASKALON. Next, we have selected the GWA as the base for the workload because GroudSim already has a loader for it and its traces are capable to represent real life scientific workloads.

Unfortunately, the GWA was focusing on the grid workloads of the past. So it is not suitable on its own to use it in a cloud context. Despite there are several workload traces already available from the cloud computing community, these traces are mostly limited to VM management operations and they rarely include VM-Task allocation information (which is an important aspect exploited in GroudSim). Also, these do not include enough information on user activities to evaluate the scalability penalties introduced in our current integration. Thus, instead of using such cloud specific workloads, we have extended the GWA traces to include virtual machine management and VM-Task allocation as described in the following paragraph.

First of all, we interpreted job submissions in the GWA trace as VM requests. The number and the kind of VMs requested were determined by the number of processors required by the particular job to be run on the VMs (e.g., if there were VM types with 1, 2, 4, 8, 16, 32 and 64 processors, and the job required 1024 processors, then we have requested 16 VMs with 64 processors). If the simulated cloud could not serve the requested VMs at once, then we applied a simple policy that retried the VM requests after some of the previous jobs have completed. Finally, upon receiving notification from the simulator that the VMs for a specific job is ready then we have allocated the job to the newly prepared VMs in such parallel fragments that filled the VMs completely (e.g., with our previous example the 1024 processor job was split to use all 16 VMs in parallel).

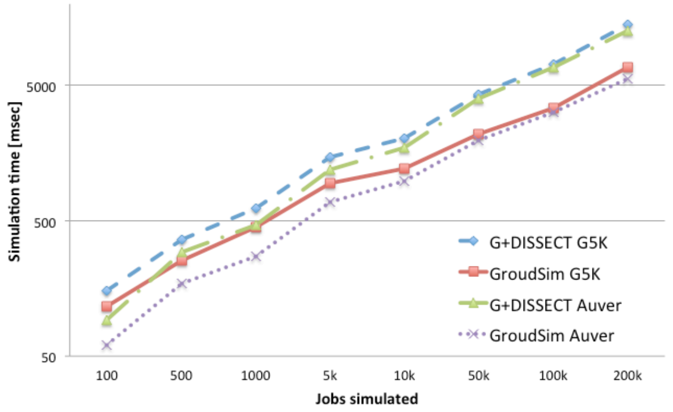

Fig. 8. Scalability analysis of GroudSim and DISSECT-CF

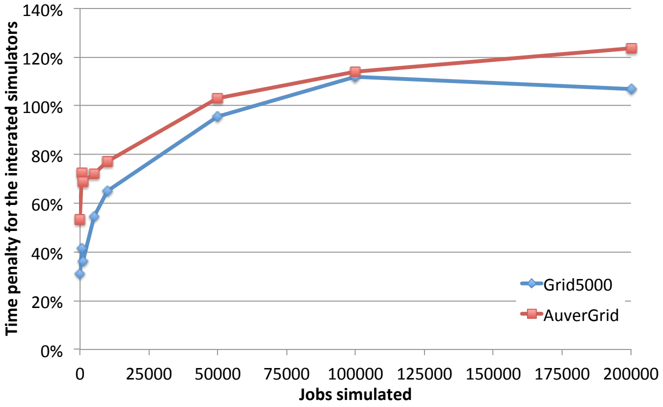

Fig. 9. Relative performance drop of the extended simulator compared to its original form

In order to analyse the effects of the integration, we have chosen to evaluate the GWA traces in both the original and the DISSECT-CF extended version of GroudSim. In both simulators, we have prepared a cloud infrastructure with a resource pool of 50 physical machines (each equipped with 64 cores, 128GBs of memory and 5TBs of disk). On this simulated infrastructure, we have executed the first N (ranging from 100 to 200.000) jobs from the Auver grid and from the Grid5000 traces in both the original and the extended simulators.

Figure 8 shows this performance analysis where one can observe a small increase in execution time is notable for both cases but the scalability of GroudSim was not harmed by the integration of DISSECT-CF. Comparing this overhead to the improved, more versatile and feature rich simulation, we concluded that this extended simulation time is still reasonable.

In terms of relative performance degradation compared to the original simulator, we have shown in Figure 9 that the

Auver grid trace simulation increased in execution time of 53% for the smallest execution and stabilised at around 120% for the biggest scenarios. Grid5000 was less affected by the integration and only had an additional execution time of 31% when simulating 100 tasks. For bigger simulations again a value around 110% was reached. The figure also shows that in smaller scale experiments, the relative performance of the simulators varies significantly (due to the internal behaviour of java virtual machines). On the other hand, after reaching around 10.000 simulated jobs, the relative performance drop of the integrated simulator stabilises and according to our measurements it never reaches 140% even with different traces than the ones we presented here.

B. New decision making opportunities in ASKALON

DISSECT-CF brings new features into the ASKALON ecosystem that will allow scientists, application developers and the ASKALON team to extend and improve their research in multiple areas and directions. In the following listing, we will introduce the research areas that open and possible with the currently integrated version of DISSECT-CF. We also provide reasons why we think these research directions are important and what we plan to research in those areas:

Resource usage. The new functionality of DISSECT-CF allows the identification of different physical machines for the instantiated virtual machines. This does not only allow to invent new methods for IaaS internal resource mapping but also in combination with workflow execution can improve the resource utilisation. In most cases, Cloud providers are seen as black boxes where the mapping of virtual machines to physical resources can only be guessed. With the integration of DISSECT-CF into GroudSim, new possibilities were opened up in ASKALON schedulers and resource managers. Knowing which instances share a physical machine can be used by the scheduler to map tasks with high data dependencies on instances that are close to each other resulting in reduced data transfer times.

New research directions can also be utilised within the IaaS provider. It is now possible to investigate different policies for physical - virtual machine mapping and their influence on performance, power consumption, utilisation and fairness. To investigate those features, internal IaaS scheduling mechanisms need to be changed which would not have been possible in GroudSim before the integration of DISSECT-CF. GroudSim had anonymous resource pools of cores only and did not understand the concept of physical machines.

Additional mechanisms might be added to better reflect real life environments. The influence of background load on the performance of virtual resources is an open research topic that can not be examined with commercial Cloud providers as their VM placement technique is not public. For security reasons, knowing about the instances of other users sharing the same physical machine are even more out of question. When a physical machine is hosting multiple instances there is always the possibility off performance loss. If the network, memory or CPU is becoming a bottleneck, then this could affect virtual machine performance in a measurable fashion. The inclusion of DISSECT-CF allows ASKALON users to elaborate techniques that detect and react to such bottleneck situations.

Power consumption. Green IT is getting more important, as power consumption and resulting CO2 emissions are becoming widely known issues to the general public. Thus, companies are starting to market their use of renewable energy or their increase in energy efficiency to attract energy-aware customers. Workflow schedulers can also offer benefits for such customers by improving scheduling and resource management through the use of DISSECT-CF provided measurements about the power draw of physical resources. The collected power measurements then allow the optimisation of workflow execution considering not only cost and time but also energy consumption. Although, contemporary cloud systems lack this metering functionality, enabling research work on the area will increase demands towards providers and prepare novel workflow management systems for times when such features become available from commercial or academic Clouds.

Multi-objective optimisation is a hot research area in the field of scheduling and with the integration of DISSECT-CF, energy consumption will be one of the most important new objectives to optimise for in the near future. Notwithstanding that resource usage frequently correlates with energy consumption, this statement can be barely true for resources offered by virtual machines. Instead, for such virtualised resources, the energy consumption is often more dependent on the utilisation of the underlying physical machine and its resources. For example, data centres can reduce the perceived energy consumption of a virtual cpu for users who under-utilise their virtual machines by allocating fractions of real cpus from the physical machines hosting these VMs. This behaviour in cloud systems could be exploited in future workflow management systems by further optimising virtual machine request patterns. In case of ASKALON, its resource manager can target virtual machine destruction requests to those VMs that are run on less efficiently utilised physical machines.

Cloud providers pay special attention to energy consumption reduction as it can directly reduce data centre operating costs. These cost reductions then can either give a competitive pricing advantage to the provider or increase its margins allowing more funds for its activities. Research in the area of VM placement is therefore not only interesting for users but also providers. Data centres could advertise that they apply environment friendly policies and users that want to support power saving would get attracted by such providers similar to renewable energy producers (which manage to sell their energy in most cases even more expensive then regular providers).

Network usage. GroudSim was developed with very little focus on network functionality as back then the focus of all workflows used in ASKALON was on the computational part. In these workflows file dependencies took only a marginal amount of data that have had to be transferred between resources. With integration of DISSECT-CF the network model of GroudSim was replaced with a more accurate one that allows more precise simulation of data intensive applications. As a result, scheduling techniques that consider data movement can exploit the more accurate file transfer predictions for such applications and can improve workflow runtimes.

Data centre configurations. DISSECT-CF allows to specify the characteristics of data centres in an easily exchangeable configuration file. Utilising this mechanism it can be evaluated what kind of data centre might be best fitting for a specific kind of workflow application. This was not possible with GroudSim as the hardware model of data centres was not existing. We aim at determining the influence of data centre configuration on workflow applications and their schedule with a series of experiments in the near future.

With the ongoing development of DISSECT-CF the supported research directions will be further extended. ASKALON users and developers will directly benefit from each new feature developed and will allow scientists to develop new methods, algorithms and solutions to cloud and workflow management related problems.

VI. CONCLUSION AND FUTURE WORKS

When evaluating scientific research, simulation tools are invaluable alternatives to real-world environments. For example in the field of scientific workflow management systems, simulators enable faster, more versatile, deterministic, and reproducible experimentation, including situations not easily reproducible in real-life. Despite their importance, current cloud workflow simulators lack sufficient support with respect to the underlying virtualised infrastructure, including energy-awareness that is highly demanded in today's data centres. To address this gap, we presented in this paper the integration of a stand-alone DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations (DISSECT-CF) with a mature real-world Cloud workflow management system called ASKALON and its underlying Grid/Cloud simulation environment called GroudSim. We discussed the challenges that appeared as the result of the originally incompatible APIs and functionalities of ASKALON, GroudSim and DISSECT-CF, and presented the required re-engineering and adjustments in three main areas: (i) event system, including event types, firing mechanisms and clock maintenance techniques, (ii) cloud representation at the level of data centre, VM and job management, and finally (iii) network construction, utilisation, sharing and organisation.

Our experimental evaluation, conducted on an over 3000 core simulated cloud infrastructure, demonstrated an improved behaviour of the ASKALON system regarding networking, energy metering, virtual machine instantiation and CPU sharing accuracy while the performance of the integrated simulators never dropped below half of the original GroudSim based simulations. We concluded that despite the improved functionality, the scalability of the simulator did not drop and was in alignment with our past results where we have shown the scaling issues in relation with simulators [7]. Finally, based on the experiments and the added new functionalities by DISSECT-CF, we have shown the kinds of advancements possible in workflow management systems that apply the newly presented simulation technique. These advancements were identified in the following fields: (i) resource utilisation improvements, (ii) power consumption optimisations for workflows and cloud providers, (iii) network aware workflow scheduling, and (iv) optimising workflow executions depending on data centre configurations.

Future work will target improvements in the GroudSim-ASKALON bridge allowing more information to be shared with the simulators regarding the executed workflows and also allowing ASKALON environment to gather more details about the simulated infrastructures. We will also focus on reducing the performance overheads caused by the duplication of some functionalities in the system (e.g., eventing) allowing GroudSim to concentrate more on the user side behaviour of clouds and grids. Finally, we plan to introduce dynamic pricing models to GroudSim by relying on DISSECT-CF's resource utilisation and energy consumption related reports.

ACKNOWLEDGMENT

The work presented in this paper has been partially supported by the Austrian Science fund project TRP 237-N23 and by the EU under the COST programme Action IC1305, 'Network for Sustainable Ultrascale Computing (NESUS).

REFERENCES

[1] I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields, *Scientific Workflows for Grids*, ser. Workflows for e-Science. Springer Verlag, 2007.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[3] K. Plankensteiner, R. Prodan, M. Janetschek, J. Montagnat, D. Rogers, I. Harvey, I. Taylor, kos Balask, and P. Kacsuk, "Fine-grain interoperability of scientific workflows in distributed computing infrastructures," *Journal of Grid Computing*, vol. 11, no. 3, pp. 429–455, 2013.

[4] A. Ahmed and A. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in *Advance Computing Conference (IACC), 2014 IEEE International*, Feb 2014, pp. 866–872.

[5] G. Sakellari and G. Loukas, "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing," *Simulation Modelling Practice and Theory*, vol. 39, pp. 92–103, 2013.

[6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011. [Online]. Available: http://dx.doi.org/10.1002/spe.995

[7] S. Ostermann, K. Plankensteiner, and R. Prodan, "Using a new event-based simulation framework for investigating resource provisioning in clouds," *Scientific Programming*, vol. 19, no. 2, pp. 161–178, 2011.

[8] A. Nuñez, J. L. Vázquez-Poletti, A. C. Caminero, J. Carretero, and I. M. Llorente, "Design of a new cloud computing simulation platform," in *Computational Science and Its Applications-ICCSA 2011*. Springer, 2011, pp. 582–593.

[9] J. D. Ullman, "Np-complete scheduling problems," *J. Comput. Syst. Sci.*, vol. 10, no. 3, pp. 384–393, Jun. 1975. [Online]. Available: http://dx.doi.org/10.1016/S0022-0000(75)80008-0

[10] S. Ostermann, K. Plankensteiner, R. Prodan, T. Fahringer, and A. Iosup, "Workflow monitoring and analysis tool for ASKALON," in *Grid and Services Evolution*, Barcelona, Spain, June 2008, pp. 73–86.

[11] G. Kecskemeti, "Dissect-cf: a simulator to foster energy-aware scheduling in infrastructure clouds," Submitted to: *Simulation Modelling Practice and Theory*, 2014.

[12] D. Rogers, I. Harvey, T. T. Huu, K. Evans, T. Glatard, I. Kallel, I. Taylor, J. Montagnat, A. Jones, and A. Harrison, "Bundle and pool architecture for multi-language, robust, scalable workflow executions," vol. 11, no. 3, pp. 457–480, Sep. 2013. [Online]. Available: http://link.springer.com/article/10.1007/s10723-013-9267-2

[13] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurrency and computation: practice and experience*, vol. 14, no. 13-15, pp. 1175–1220, 2002.

[14] S. Murugesan, "Harnessing green it: Principles and practices," *IT Professional*, vol. 10, no. 1, pp. 24–33, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1109/MITP.2008.10

[15] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," in *E-Science (e-Science), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 1–8.

[16] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899 – 2917, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731514001105

[17] A. Sulistio, U. Cibej, S. Venugopal, B. Robic, and R. Buyya, "A toolkit for modelling and simulating data Grids: an extension to GridSim," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 13, pp. 1591–1609, 2008.