An agent based network resource planner for workflow applications

Zhiming Zhao Paola Grosso Jeroen van der Ham
Ralph Koning Cees de Laat
System and Network Engineering research group
Informatics Institute, University of Amsterdam
Science Park 904, 1098XH, Amsterdam, the Netherlands
{Z.Zhao|P.Grosso|vdHam|R.Koning|C.T.A.M.Delaat}@uva.nl

November 7, 2011

Abstract

Many scientific workflow applications are driven by simulation generated data, or data collected from sensors or instruments, and the processing of the data is commonly done at a different location from where the data is stored. Moving large quantities of data among different locations is thus a frequently invoked process in scientific workflow applications. These data transfers often have high quality requirements on the network services, especially when the application requires steering from human interaction. Advanced networks such as hybrid networks make it feasible for high level applications to request network paths and service provisioning. However, current workflow applications tune the execution quality neglecting network resources, and by selecting only optimal software services and computing resources. Including network services in the resource scheduling adds an extra dimension for workflow applications to optimize the runtime performance. In this paper we present a system called NEtwork aware Workflow QoS Planner (NEWQoSPlanner) to complement existing workflow systems on selecting network resources in the context of workflow composition, scheduling and execution when advanced network services are available.

1 Introduction

Many scientific workflow applications are driven by simulation generated data, or data collected from sensors or instruments, and the processing of the data is commonly done at a different location from where the data is stored. Moving large quantities of data is thus a frequently invoked process. In many cases these transfers have high quality requirements on the network services, especially when the application requires steering from human interaction.

The statistical performance guarantees what a network system can make in terms of throughput, delay, and packet loss, namely network Quality of Service (QoS), essentially determines the upper bound of the performance that data delivery can achieve. Tuning the quality of the network service plays an important role in the optimization of the application performance; however, current workflow applications tune the execution quality neglecting network resources, and by selecting only optimal software services and computing resources. There are several reasons: 1) traditional IP-based networks provide limited reservation capabilities for workflow engines; 2) the existing e-Science applications assume available network connections are non-changeable services, and seek customized solutions at software level to optimize computing processes and data storage; and 3) the existing applications mainly consider the functionality of the e-Science services, and provide limited support for including (network) quality requirements for the services in the composition, enactment and execution of workflows.

Several strategies have been tried to improve the data movement performance without manipulating the network service quality, such as caching data closer to the computing element [1], or reducing the load of computing tasks by reusing the previous computed results [2]. However, for applications that require data streams from remote sensors, such as in [3], those solutions will not be sufficient to assure the global workflow quality requirements without guarantees at the network service level.

The recent emergence of advanced network infrastructures for e-Science enables tuning of network performance at the application level. The idea of providing data intensive applications with deterministic point-to-point connections was fostered by a community of research networks, later organized in the Global Lambda Integrated Facility (GLIF). A lambda is a light wavelength used to carry information in optical systems. This community provides a global network to support data-intensive scientific research, and also supports middleware development for optical networking. The ideas in this community led to the concept of hybrid networking, the offering of packet switched (IP) services and circuit switched connections over the same physical network infrastructure [4]. In the meantime, the available speeds of lightpaths keep growing. A lightpath is a network circuit transmitting data between two end points using a specific wavelength. While 10Gbit/sec links were introduced only a few years ago, 40Gbit/sec and 100Gbit/sec links are now becoming available to application developers [5]. These kinds of links provide unique opportunities for transport of high-quality media. By including network resources in the scheduling loop, the high level application gets an extra opportunity to optimize execution and improve performance.

These new opportunities come with some challenges. Not all network infrastructures provide the network services for reserving specific connections or allocating network bandwidth; the service invocation in different network domains is often proprietary and not easily extensible, and makes a request for network service provisioning across sites difficult [6]. Scheduling network resources requires knowledge on the current state of the network, which implies the existence of a sophisticated monitoring system. The solutions to these issues require not only the integration among information sources from distributed infrastructure domains, but also the intelligence to invoke different levels of network and application control services. Agent technologies provide a suitable model to decompose and encapsulate system functionality and to implement the interaction among the system components.

In this paper, we discuss the challenges in including network resources in scientific workflows, and present an agent based architecture to accomplish this goal and realize application level quality tuning of network resources. We apply this in the context of an ongoing project, CineGrid [7]. The paper is organized as follows; first, we will review the state of the art in this field, and then present our solution and the system prototype. The preliminary description of the system architecture has also been presented in the workshop on Agent Based Computing 2010.

2 Research background

Agent technologies play an important role in implementing workflow system components and the workflow execution control. In the following sections, we first briefly review the usage of agent technologies in workflow systems, and then review how QoS of the resources are included in the workflow life-cycle to optimize system performance.

2.1 Agent technologies in workflow systems

The Agent Oriented (AO) methodology complements the object and component oriented methods with knowledge related notions to manage system complexity [8], and emerges as an important modelling and engineering approach for constructing complex systems, such as workflow management systems. During the past decades, agent based models have been applied as an advanced technology in workflow systems for wrapping functional components [9], controlling the execution and realising cooperative computing among distributed processes. Czarnul et al., presented a multi agent based cooperative model, namely BeesyBees, to schedule the computing tasks of distributed workflow processes [10]. Zhao et al., [11] presented an agent based software bus to couple heterogeneous workflow engines in one meta workflow application. Agent frameworks, such as FIPA [12], abstract the structure of basic agents and define standardized communication languages to represent interactions between agents, which facilitate the implementation of agent based applications. The agent communication language (ACL) provided by the FIPA frameworks introduces ontology based schemas to wrap communication messages between agents; it provides a suitable solution to integrate semantic information provided by the data sources and agents.

2.2 Resources QoS in Scientific workflows

Workflow systems have been recognized as an important tool to manage the invocation of lower level network services and the security related routines [13]. In SC09, the VLAM workflow system [14] demonstrated allocation of workflow modules over specialized network connections. Several e-Science environments already recognized the importance of network resource reservation in the context of workflow scheduling, for instance in [15]. From the life-cycle perspective of a scientific workflow, QoS is relevant to three aspects: workflow composition, execution control and provenance. We briefly review how resource QoS is included in each phase of the life-cycle.

The QoS-awareness of *workflow composition* mainly refers to selecting optimal services and composing a workflow of the highest quality. The problem of service selection has been formulated differently. A commonly used formulation is *shortest path finding in a weighted graph*, in which the available services are represented as a directed graph according to the service types, and the graph nodes are labelled by the quality attributes of the service [16]. Well known shortest path finding algorithms include Bellman-Ford and Dijkstra's. These algorithms exhibit optimal performance because of their greedy search strategy and the avoidance of backtracking operations during the search; however, the minimal cost path found by the algorithms is often not the most optimal solution if there are extra constraints on the quality attributes. Therefore, the problem has also been formulated as a multi constraint optimal path problem [17], or multi objective optimization problem. Ant colony optimization (ACO) is a meta heuristic search approach proposed in [18] for discovering low cost paths in a graph, and for solving NP-hard combinatorial optimization problems. Fang et al, [19] applied ACO in service selection and proposed a multi objective ACO approach which can simultaneously optimize several objectives. Genetic algorithm in searching optimal paths, and constraint or Integer programming methods are also widely used for the multi objective optimization problem.

Workflow execution is the mapping of workflow processes to underlying computing resources and the scheduling of the execution sequence. Task based scheduling is a straightforward approach, in which the workflow tasks are submitted to the local manager of the computing infrastructure. Several researchers have instead proposed a workflow level scheduling that takes into account future task performance [20]; this approach will achieve higher performance and better resource utilization than only using local resource managers. Multi objective optimizations are widely used to formulate the problem of QoS aware scheduling. Avanes proposed a constraint programming based approach to search for best match between workflow requirements and the available computing resources [21]. The basic idea is to describe the quality requirements and resource dependencies as constraints by partitioning the workflow into different parts based on its patterns and their QoS requirements. One of the contributions from Avanes work is that the network dynamics has been also included in the procedure of constraint resolving. Resource provisioning plays an important role to improve the fault tolerance and the performance of the workflow [22]. Basically, provisioning can be either static or dynamic. Advanced reservation is a typical static provisioning mechanism, and several batch based schedulers support it. Based on the high level quality requirements, the workflow engine reserves computing resources and time slots from the Grid resource manager. One of the disadvantages of static provisioning is its overhead on the total cost for computing the workflow. To improve this, Raicu et al. [23] proposed multi level scheduling strategies, in which the application level scheduler is able to interact with the low level resource manager to tune the requirements at runtime. This approach introduces a dynamic component in the provisioning process.

The *provenance* service tracks the events that occurred in the workflow execution, and allows scientists to trace the evolution of data computed in the workflow and to obtain insights in the experiment processes. Moreover, provenance data can also be used to debug errors of the workflow execution and optimize the workflow design. The Open Provenance Model (OPM) [24] emerges as a standard model to represent workflow provenance information. Including QoS information of workflow processes and the execution in the provenance model allows scientists to analyze the quality of the services and the workflow scheduling. In [25], the provenance service is provided by a QoS aware middleware, which records the changes of the service quality as events. Evaluating trust and reliability of the provenance data itself has also been discussed in the literature [26]. However, research on the provenance model that includes the QoS information of the workflow processes is still in its very early stage.

The above technologies contribute the necessary building blocks to deliver large quantities data between e-Science application components. However, putting them all together and providing quality guarantees for the whole application in terms of high quality of both data and its delivery is not trivial; besides the fact that network QoS is not directly included in the scheduling loop, network QoS also requires monitoring of the infrastructure, which is not an easy task.

2.3 Research motivation and problem descriptions

Our research interest focuses on the inclusion of the network quality of service in the high level e-Science workflows. In this context, we will investigate 1) how to describe the application level requirements and map them onto the quality model of underlying network resources, 2) how to select network resources and to compose them in the high level workflow processes, and 3) how to utilize network services in the workflow application at run time. We propose a QoS aware planner which covers the life-cycle of workflows not only at composition, but also at the execution phase.

3 Network aware workflow QoS planning

In the previous section, we reviewed different technologies involved in delivering scientific data over networks, and explained that including network QoS in high level applications is essential to add global quality guarantees on an application. In this section, we will discuss an agent based solution for this problem. Our focus is on improving the existing workflow systems by adding an extra planner. We had two alternatives when we looked at the inclusion of QoS aware functionalities in a scientific workflow system: 1) re-engineer the functional components of an existing workflow system to include the QoS support, or 2) consider the existing workflow systems as legacy systems, and provide QoS support as plugged components to the system. Each alternative exhibits advantages and disadvantages. In the context of our research, we chose the second approach; one of the motivations is that generic functional components can be encapsulated as reusable tools which can serve different specific scientific workflow systems to get QoS support.

3.1 Design requirements

We can highlight three scenarios where network QoS support can be applied: QoS aware resource selection, resource provisioning and quality assured workflow execution. The designed system thus needs to meet the following functional requirements:

1. The system must include QoS aware resource discovery and selection of network resources. Network resources and the quality attributes are necessarily described, and a search tool is provided to check the suitable resources based on the input requirements.

2. The system should be able to generate a resource provisioning plan for selected resources based on the input requirements. The plan is made based on provisioning services provided by the available network infrastructure.

3. The system should be able to generate workflows handling large data movement between network resources with guaranteed data transfer quality, and wrap the generated workflow as a service which can be executed standalone or included in a third party workflow. At runtime,

the system should provide monitoring services to track the actual quality of the network resources. It should also provide interfaces that can be invoked by third party workflows during their provenance procedure to record all the runtime information.

3.2 An agent based network QoS planner

Using the agent based modelling, we propose an architecture, called NEtwork aware Workflow QoS Planner (NEWQoSPlanner), for including the network level quality control in available workflow systems. The NEWQoSPlanner system is able to 1) select optimal network paths between workflow processes which need to transfer large quantities of data, and 2) manage the network level operations, such as provisioning and establishing links, for the high level workflow system via a proper invocation interface, such as a service or a compatible description of high level workflow.

The NEWQoSPlanner architecture consists of six agents: *a Resource Discovery Agent (RDA), a Workflow Composition Agent (WCA), a Resource Provisioning Planner (RPP), a QoS Monitor Agent (QMA), a Provenance Service Agent (PSA)*, and a coordination agent called *QoS aware workflow planner (QoSWP)*. Figure 1 illustrates a conceptual schema of our agent system.

[Figure 1 about here.]

The QoSWP coordinates the other agents to select suitable services, to propose optimal network connections between the services, and to create the necessary scripts for the workflow engine to invoke the requested services. A typical use case scenario will illustrate the role of each component (see Figure 1). The QoSWP receives the request for data processing services and the service requirements from the user (step1). After that, the RDA reads the description of the resources and the network topologies from the registry, and searches suitable data sources and destinations, and network paths between them (step2). The RDA returns a list of qualified candidates, and sorts them based on the quality metrics of each candidate (step3). From the candidates, the QoSWP selects the best one, and requests WCA and RPP to generate a resource provisioning plan and a data transfer workflow (step4 and step5), both of which will be executed by the workflow engine (step6). At run time, the QMA monitors the actual state of the resources and checks whether the global quality required by the workflow is satisfied (step7). Based on the states updated by the QMA, the QoSWP decides whether the resources of the workflow should be adapted. The provenance service records events in the provisioning and allocation of resources, and combines the actual state of the quality attributes with the log data (step7).

3.3 Design considerations

The functional components of NEWQoSPlanner are wrapped as agents using the agent middleware. JADE (Java Agent DEvelopment Framework) is a free software package and distributed by Telecom Italy [12] and provides a suitable solution to develop the system. Fully implemented in java, JADE realizes a FIPA compliant multi agent middleware. In our project, a number of reasons motivate us to choose JADE as the implementation framework. First, the JADE platform can be distributed across machines and the configuration can be controlled via a remote GUI. The Java language makes the development portable; the JADE framework allows agents move from one machine to another at runtime. Moreover, being compliant to the FIPA protocol, JADE provides a standard architecture for scheduling agent activities, which makes the inclusion of high level functionality easy, e.g., adding a Prolog module for activity reasoning. Finally, the ontology enabled agent communication between agents promotes seamless integration between the semantic network description, QoS aware searching modules, underlying models of workflow descriptions, and other necessary functional components of our system.

In the NEWQoSPlanner system, RDA is one of the core components; it provides information for further resource provisioning (RPP), and the creation of an invocation list of network services (WCA). Semantic descriptions of the network resources and the searching technologies that allow workflow systems to discover proper network resources are the basis for realising the RDA. In the rest of the section, we will focus on the implementation of RDA.

3.4 Modelling quality constraints in abstract workflow processes

To propose network resources to high level workflow applications, a proper schema that workflow systems can specify requests on data movement is needed. In the system, RDA selects network paths based on descriptions of stack of resources which range from network topology, devices, computing and storage services, and data content provided by those storage services. In this sub section and the next we will discuss these two issues respectively.

The input to the RDA is the requirements for data related processes which are needed by the high level workflow. Based on the experience of early work [27–29], we propose an ontology for describing abstract workflows processes (*qosawf.owl*). It defines the basic concepts of workflow processes, pre/post/execution conditions of the process, media data, and quality attributes, as shown in Fig 2.

[Figure 2 about here.]

We abstract several basic data related *Processes*: *Archive_data*, *Move_data*, *Playback_data*, *Transformation*, and *Browse*. A *Process* class uses *pre_Condition* and *post_Condition* to indicate the requirements for *Data* the process requires and generates, and the quality for the required data. The *Process* class also uses *execution_Condition* to indicate the service quality for the process. The description of the user request is thus described as an object of the *Request* class, and a *Request* consists of one or more *Processes* which can be accessed via the *request_Functionality* property. In the current definition, the *Data* has two specific types: *Media* and *Scientific_Data*. And the service quality is modelled as set of *Quality_Attributes*. Based on the QoS taxonomy defined in [30], *Quality_attribute* can more specifically be *Precision*, *Timeliness*, *Reliability* and *Security_Level*. In our case, where the pre and post conditions consist of requirements for data and the data quality, *and_Condition* and *or_Condition* are the two most important types.

3.5 Semantic description of network resources

The semantic web technologies provide suitable solutions to describe network topologies, devices, and the QoS requirements for data services. In the CineGrid project, semantic web technologies are used to describe the services, devices and the network topology. The abstract workflow ontology is linked with ontologies for describing lower level CineGrid resources using special properties. We have developed ontologies for describing network topologies, and services and data that are on top of the network infrastructure respectively.

The Network Description Language (NDL) [31] models the different levels of a network infrastructure: physical, domain, layer and topology[1]. NDL contains a modular set of schemata, defining an ontology to describe computer networks. The topology schema describes devices, interfaces and connections between them on a single layer. The classes and properties in the topology schema describe the topology of a hybrid network, without detailed information on the technical aspects of the connections and their operating layer. The layer schema describes generic properties of network technologies, and the relation between network layers. The topology schema defines network topologies on a single layer. The NDL layer schema allows applications to describe multi-layer networks, like hybrid networks. The domain schema describes administrative domains, services within a domain, and how to give an aggregated view of the network in a domain. It allows network operators to provide an aggregated view of their domain to neighbouring domains, rather than the full topology. And the CineGrid Description Language (CDL) [7] describes the devices on top of the network infrastructure, and services provided by those devices. The devices include visualization related equipments, such as projectors and displayers, and computing and storage related equipments. And services provided by the devices include data streaming, visualisation, and storage [2].

Owl provides three build-in properties to map ontologies: *owl:sameAs* between instances, *owl:equivalentClass* between classes, and *owl:equivalentProperty* between properties. The CineGrid resources are integrated with the network level resources via the property *owl:sameAs*. The mapping between abstract workflow and the CineGrid resources is via the property: *qosawf:implemented_By*, which contains sub properties for each specific process. The process *qosawf:Playback_Data* is linked to the service defined in CDL *cdl:Visualizer* via *qosawf:playbackData_Implemented_By*. To decouple the abstract workflow from the resource descriptions, the mapping classes from *qosawf* and *CDL* are defined in a separated file *qosawf-ontmap-cdl.owl*, which is loaded by the reasoning component when it is sure the underlying resource is using CDL. In future, when a new resource description ontology is introduced, the change can only be made in the ontology mapping file.

[1] Available at: http://cinegrid.uvalight.nl/owl/ndl-domain.owl and http://cinegrid.uvalight.nl/owl/ndl-topology.owl

[2] Available at: http://cinegrid.uvalight.nl/owl/cdl/2.0

3.6 Resource discovery

After comparing different Query languages (RQL [32], RDQL [33], SeRQL [34], SPARQL [35]) and Rule languages (SWRL [36], Prolog/RDF lib [37], JESS [38] etc.), we chose the semantic web library of SWI-Prolog to implement the searching component of the RDA. The triple based manipulation interface of the semantic web library of SwiProlog is easy for the high level language to access the runtime state of the triples. Moreover, the Prolog language provides effective solutions to realise path findings in a graph. After being loaded in the Prolog database, the triples defined in the ontology file can then be accessed and queried via the term *rdf(subject, predict, object).*

To select network resources, the resource discovery agent 1) parses the input description, 2) searches suitable CineGrid resources which meet the requirements of the data sources and destination, 3) looks for optimal network paths between them, and 4) computes the quality of resource candidates and proposes solutions.

Abstract workflow process parsing. The input of RDA contains functional requirements for data operation (*Process*) and the quality requirement for both the operation and the data. The current QoSAWF schema allows one input description to contain only one instance a *Process* concept.

The parsing procedure obtains the pre/execution/post condition of the process. The *pre_Condition* and *post_Condition* of a process contains both requirements for data, such as data type and properties, and for the quality of the data, such as resolution if the data is a video file. The *execution_Condition* gives QoS requirements for the process. The RDA selects an element in the pre/execution/post condition, and uses it as the constraints for the resource search.

Search data and the operation. From the data requirements derived from the step 1, the host which contains the required data, namely *data sources*, and the host which will consume or store the data, namely *data destination*, can then be located. From the resource description, the RDA derives the set of storage services which contain the *Data* instance that meets the required type, and quality. In the CineGrid, each *Data* instance is associated with a *Meta_data* object, which can be accessed via the property *hasMetadata*. Therefore, the sources of data are located by searching instances of *Data* which contain meta data that meets the requirements abstracted from the *pre_Condition*. Using the property of *cdl:provided_by* and *owl:sameAs*, the actual host that stores data can then be derived.

The destination of the data is derived from the process types described in the requirement. As we mentioned above, based on the type of data operations, we abstract three basic process types: *MoveData*, *PlayData*, and *ArchiveData*. For the process of *PlayData*, *post_Condition* can be empty, because the process does not generate data. The processes are linked to the actual services of CineGrid via property *implemented_By*. Therefore, the location that data processing will take place is determined by both the location of the implemented services and the location of the data required in the *post_condition*.

Network paths finding. The next step is to find all network paths between the data sources and destinations. Using the ontology of NDL, a network path can be found using three properties: *link_to*, *connect_to* and *switch_to*. The *link_to* property indicates that two network devices are directly connected via a physical line, while *connect_to* refers to a connection which might include unknown devices between the two end points of the path. The *connect_to* property is mostly used in the situation where two devices belong to two different domains and the detailed physical connections between them is not clear or not open to public due to administration rules. The *switch_to* property is only used in a switch device to indicate the connectivity between different ports in the device.

The rdf triples defined in the network topology description give suitable *graph* representation for finding network paths.

Rank the candidates. The first three steps return resource candidates which are represented as (source, destination, path). The quality of the resource can be evaluated at three levels: 1) the quality of data, 2) the quality of the storage/stream services, 3) quality of the hosts which provide the services, and 4) the network connection between hosts.

From the CDL and NDL ontology, the RDA can abstract the following quality attributes: 1) the quality of data, such as compressed ratio and resolution, from the data catalogue of resources 2) the properties of host, such as its CPU speed, memory size and the available storage space, 3) the network bandwidth of network connections. From these quality information, and the quality requirements defined for the process, the RDA applies the following rules to filter unqualified candidates from the searched results:

1. The RDA first checks if the data and services meet the quality requirement;

2. Then compute the bandwidth of the candidate network path, only the

candidates that have bandwidth meeting the minimal data transfer rate are kept;

3. The RDA sorts all qualified candidates based on the quality of the host which provides data or process service, and the bandwidth of the network connections.

3.7 Utilizing resources at runtime

The ultimate goal of our NEWQoSPlanner is that it can automatically find a workflow for processing media and provide a quality guaranteed network path through the network. One of the final steps in this process is to provision a path in the network that becomes available to the executing workflow. So far we have only integrated the NEWQoSPlanner in an ad hoc fashion with our network test bed. The planner can execute some scripts to create network paths in our experimental network. Obviously such an approach does not scale to larger networks. There would be problems with authentication and authorization, supporting different kinds of network equipment, compatibility with other systems, et cetera. An easier solution is to integrate with existing network management tools. This will allow the workflows to be used for intra-domain path selection in many more networks, or even inter-domain using the global GLIF network.

There are currently several network provisioning systems that allow integration with other applications. ESnet and Internet2, two large research and education networks in the USA, have developed the On-Demand Secure Circuits and Advanced Reservation System (OSCARS) [39]. This system allows users to create reservations for circuits in the ESnet and Internet2 network. The system can use either MPLS and RSVP to create connections, in the case of the ESnet network, or integrate with Internet2's Dynamic Circuit Network and provision VLANS on their national backbone network. The OSCARS system allows users to specify different properties that a circuit reservation should fulfill, such as bandwidth, or a specific VLAN number. The OSCARS system also allows applications to use the WebService interface for a more direct provisioning service. This kind of integration would be ideal for our NEWQoSPlanner.

Another system currently available is the OpenDRAC system [40], originally developed by Nortel Networks. This provisioning system is currently in use on the SURFnet network in the Netherlands. The management system provides the network operator with the tools to manage and monitor the network, but also has an interface for users to request lightpaths. Depending on the access rights of the user he can request lightpaths from several locations with different capacities. OpenDRAC allows users to specify other attributes of the circuit as well, such as bandwidth, VLAN ID, etc., depending on the capabilities of the underlying network. The OpenDRAC system also features a WebService interface, which allows for simple integration with other applications.

Currently, the different management systems such as OpenDRAC and OSCARS are not directly compatible, meaning that it is not possible to create a reservation which goes from a domain managed by OpenDRAC to an OSCARS managed domain or vice-versa. There currently is a demonstration project going, called Fenius [41], to implement a simple inter-domain interface between these provisioning systems to allow for the automatic set up of inter-domain circuits. This has been demonstrated successfully at the SuperComputing 2010 conference. In the future this will converge to a standard currently in development in the Open Grid Forum (OGF), called the Network Service Interface [42]. This standard will allow provisioning systems to interact with each other to automatically create inter-domain circuits for the users and their applications.

4 A use case and performance characteristics

The current system is prototyped using the JADE framework. The planner can be invoked via either an Agent Communication Language (ACL) based interface by other FIPA compliant agents, or a Web service interface by workflow applications. In this section, we will first demonstrate the functionality of current prototype using a CineGrid example, and then investigate its performance characteristics.

4.1 A CineGrid use case

We take a use case from the CineGrid project to demonstrate the functionality of Resource Discovery Agent (RDA) provided by the NEWQoSPlanner. In the use case, a user can browse and select multimedia material from the CineGrid environment via a web portal, and perform actions like preview, playback or backup on the media material. A user can describe not only the name and properties of the media, but also the quality requirements for actions on the media. The portal encapsulates the user input using the QoSAWF schema, and sends it to the RDA as a request via a web service interface. The RDA looks for all possible network paths between the sources (storages contain the media material) and the destination (visualization or storage) devices where operations may take place. The portal visualizes the searching results, and allows the user to select suitable candidate from the list.

The portal currently supports two types of interaction: 1) browse and select the media content, 2) compose the requirements using the options provided by the portal interface. For instance, a user can ask for *playing* a movie with name *Bridge* on a 4K resolution (*4000×3000* pixels) screen with frame rate 30 per second. The up part of Figure 3 shows the visual diagram of the request. The bottom part of Figure 3 also shows a screen snapshot of the portal and the network topology of the test bed.

[Figure 3 about here.]

In the experiment, we deploy the RDA in a Linux host with 4 core and 8G memory, and the portal is launched on the same host. The overall delay between portal and the search engine is averagely between 0.006 to 0.4 seconds, depends on the types of interactions. In the next section, we will investigate more performance characteristics of the current prototype.

4.2 Performance characteristics

The performance of the RDA can be quantified as the time cost for finding data sources, destinations and network paths for a given request (see section 3.6). In this section, we will first measure the time costs for each part, and then investigate how the network topologies influence the performance.

In the experiments, we create several resource descriptions with diverse topologies and scales to simulate different types of infrastructure. The network topologies are modelled using the connections between switches in the network. To keep the experiments simple, the artificial descriptions only contain two types of devices: switches and hosts. The switches are connected as random trees, and hosts are randomly connected with those switches. All the ports in the switches are fully connected, and only static interfaces are include. We create descriptions with different number of switches, each switch by default has 48 ports, and each host has one interface. For each configuration, we create its network descriptions (NDL compliant), service descriptions (CDL compliant) and data descriptions. To keep the focus on the network part, we do not change the scale of the data descriptions.

We run the request showed in Figure 3 on different test bed descriptions we created. The descriptions range from a simple test bed with one switch and 24 hosts to 32 switches and 768 hosts. Figure 4 shows the average of 16 measurements, and the error bars are the standard deviations.We can clearly see that the time cost for path finding increases when the network topology increases.

[Figure 4 about here.]

Since we did not change the scale of data description files when creating experimental test bed descriptions, the time cost for finding data sources thus remains consistent as shown in Figure 4. However, the scale of service descriptions changes while the network scale increases, which also makes the time cost for finding data destinations increase as shown in the figure.

The RDA currently finds a network path between a given source and destination node using a Depth-First strategy implemented in Prolog. To find all data sources, destinations and paths, the *setof* command provides a straightforward solution in Prolog to search all candidates; however, we find such implementation can cause bad performance when the scale of the network increases (See Figure 4). One of the reasons is that the large number of backtracking occurred during searching procedure are resource and time consuming. Moreover, it is not easy to add extra stop condition to the *setof* command to limit the searching space. To improve it, we combine the Breadth-First strategy in the system to optimize the searching procedure for more network paths. Figure 5 shows the comparison between two strategies and the time cost for finding only one candidate. We can see a big performance improvement. Currently, the RDA only deals with one layer network, the more sophisticated multi layer network path finding algorithms [43] developed in our group will be included in the next implementations.

[Figure 5 about here.]

4.3 Discussion

To include network resources in the workflow systems, our system tackles the problem from a top down approach. We do not aim at integrating a workflow system with a specific network infrastructure as presented in [14]. We highlight the quality requirements of the data movement processes from high level scientific workflows, model them as a lightweight schema called QoSAWF, and map them onto the semantic descriptions of stack of resources, such as data, services and network infrastructure. In the experiment, we have demonstrated that the QoSAWF schema is capable of describing the interaction scenarios required by the CineGrid use case. Compared to the network resource discovery in the Mobile Ad-hoc NETworks (MANETs) [44], our approach more focuses on the circuit network and on the large data movements, although both work share changing network routings behaviour as the common objective.

In the use case, the candidates found by the planner will be finally selected by the user. In order to integrate with a workflow system, the planner still has to solve several issues. First, the NEWQoSPlaner has to employ a resource cost model to decide which candidate is the most cost-effective for workflow process and then it should also provide the composition of the temporal workflow for reserving and invoking network resources. These issues are certainly in the agenda of our future work. Nevertheless, as the basis for the NEWQoSPlanner system, the network resource search component is toward the direction.

As we mentioned in 3.6, a request currently only allows one data action process; for a workflow which has multi process need network quality support, the network resources will be searched separately for each process. This strategy works for simple cases; however, if the workflow processes are dependent and the network resources for supporting the data operations in those processes can be optimized at a higher level. Therefore, a multi data process search will also be needed. This will certainly be an issue in our future research.

5 Conclusions

In this paper, we presented our work, namely the NEWQoSPlanner agent framework, on including network resources in scientific workflows. We developed an ontology based description schema for QoS aware requirements, and discussed implementation details of the searching mechanism of network resources.

From the use case and experiments, we can at least draw the following conclusions.

1. For the scientific workflows in which large data movement is the performance bottleneck, quality control at the network level is crucial;

2. Semantic technologies play an important role in modelling QoS attributes and mapping quality description between different layers of resources in workflow system;

3. The QoSAWF ontology provides a light solution to describing QoS requirements for data operation related workflow process;

4. The NEWQoSPlanner provides agents for searching suitable network resources and is towards the direction of network quality adaptive planner workflow processes.

6 Future directions

In addition to the future work we enumerated in the section 4.3, another important issue in our research agenda is to develop a suitable semantic model for logging and querying workflow processes with the network QoS information. With this model, namely provenance model, the runtime information of the workflow and the network events will be recorded for the further querying for reproducing execution scenarios of the workflow and for investigating the dynamics of network services in the workflow execution.

Acknowledgment

We would like to thank the Amsterdam e-Science Institute and the Dutch national and education network SURFnet, through the GigaPort Research on Network (RoN) project for sponsoring this research.

References

[1] David Chiu and Gagan Agrawal. Hierarchical caches for grid workflows. *Cluster Computing and the Grid, IEEE International Symposium on*, 0:228–235, 2009.

[2] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Scott Koranda, Albert Lazzarini, Gaurang Mehta, Maria Alessandra Papa, and Karan Vahi. Pegasus and the pulsar search: From metadata to execution on the grid. In *PPAM*, pages 821–830, 2003.

[3] Robert Morris, Jennifer Dungan, and Petr Votava. A workflow model for earth observation sensor webs. *Space Mission Challenges for Information Technology, IEEE International Conference on*, 0:69–76, 2009.

[4] C. Laat, E. Radius, and S. Wallace. The rationale of the current optical networking initiatives. *Future Generation Computer Systems, 3rd Biennial International Grid Applications-Driven Testbed Event*, 19(6):9991008, 2003.

[5] C. Dumitru C., F. Koning, and C. Laat. Clearstream: End-to-end ultra fast transmission over a wide area 40gbit/s lambda. In *Demo Supercomputing 2010*, 2010.

[6] Zhenhai Duan, Zhi-Li Zhang, and Yiwei Thomas Hou. Service overlay networks: Slas, qos and bandwidth provisioning. *Network Protocols, IEEE International Conference on*, 0:334, 2002.

[7] Cinegrid initiative, http://www.cinegrid.org.

[8] Philippe Massonet, Yves Deville, and Cèdric Néve. From aose methodology to agent implementation. In *Proceedings of the first international joint conference on Autonomous agents and multi agent systems*, pages 27–34. ACM Press, 2002.

[9] Li Guo, Moustafa Ghanem, Vasa Curcin, and Nabeel Azam. Intelligent agents in the service-oriented world - an industrial experience report. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 1:693–696, 2009.

[10] P. Czarnul, M. Matuszek, M. Wojcik, and K. Zalewski. Beesybees-agent-based, adaptive & learning workflow execution module for beesycluster. In *Information Technology (ICIT), 2010 2nd International Conference on*, pages 139 –142, june.

[11] Zhiming Zhao, Suresh Booms, Adam Belloum, Cees de Laat, and Bob Hertzberger. Vle-wfbus: a scientific workflow bus for multi e-science

domains. In *Proceedings of the 2nd IEEE International conference on e-Science and Grid computing*, pages 11–19, Amsterdam, the Netherlands, December 4- December 6 2006. IEEE Computer Society Press.

[12] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE: a FIPA2000 compliant agent development environment. In *Proceedings of the fifth international conference on Autonomous agents*, pages 216–217. ACM Press, 2001.

[13] Y. Demchenko, L. Gommans, C. de Laat, A. Taal, A. Wan, and O. Mulmo. Using workflow for dynamic security context management in grid-based applications. *Grid Computing, IEEE/ACM International Workshop on*, 0:72–79, 2006.

[14] Adam S. Z. Belloum, David L. Groep, Zeger W. Hendrikse, Bob L. O. Hertzberger, Vladimir Korkhov, Cees T. A. M. de Laat, and Dmitry Vasunin. Vlam-g: a grid-based virtual laboratory. *Future Gener. Comput. Syst.*, 19(2):209–217, 2003.

[15] Christoph Barz, Markus Pilz, and Andr Wichmann. Temporal routing metrics for networks with advance reservations. *Cluster Computing and the Grid, IEEE International Symposium on*, 0:710–715, 2008.

[16] Yingqiu Li, Minghua Chen, Tao Wen, and Lei Sun. Quality driven web services composition based on an extended layered graph. *Computer Science and Software Engineering, International Conference on*, 3:153–156, 2008.

[17] Jia Yu, Michael Kirley, and Rajkumar Buyya. Multi-objective planning for workflow execution on grids. *Grid Computing, IEEE/ACM International Workshop on*, 0:10–17, 2007.

[18] In Alaya, Christine Solnon, and Khaled Ghdira. Ant colony optimization for multi-objective optimization problems. *Tools with Artificial Intelligence, IEEE International Conference on*, 1:450–457, 2007.

[19] Fang Qiqing, Peng Xiaoming, Liu Qinghua, and Hu Yahui. A global qos optimizing web services selection algorithm based on moaco for dynamic web service composition. *Information Technology and Applications, International Forum on*, 1:37–42, 2009.

[20] Fumiko Harada, Toshimitsu Ushio, and Yukikazu Nakamoto. Adaptive resource allocation control for fair qos management. *IEEE Transactions on Computers*, 56:344–357, 2007.

[21] Artin Avanes and Johann-Christoph Freytag. Adaptive workflow scheduling under resource allocation constraints and network dynamics. *Proc. VLDB Endow.*, 1(2):1631–1637, 2008.

[22] Gideon Juve and Ewa Deelman. Resource provisioning options for large-scale scientific workflows. In *ESCIENCE '08: Proceedings of the 2008 Fourth IEEE International Conference on eScience*, pages 608–613, Washington, DC, USA, 2008. IEEE Computer Society.

[23] Ioan Raicu, Yong Zhao, Catalin Dumitrescu, Ian Foster, and Mike Wilde. Falkon: a fast and light-weight task execution framework. In *SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, pages 1–12, New York, NY, USA, 2007. ACM.

[24] Luc Moreau, Juliana Freire, Joe Futrelle, Robert E. Mcgrath, Jim Myers, and Patrick Paulson. The open provenance model: An overview. pages 323–326, 2008.

[25] Anton Michlmayr, Florian Rosenberg, Philipp Leitner, and Schahram Dustdar. Service provenance in qos-aware web service runtimes. *Web Services, IEEE International Conference on*, 0:115–122, 2009.

[26] Shrija Rajbhandari, Arnaud Contes, Omer F. Rana, Vikas Deora, and Ian Wootten. Trust assessment using provenance in service oriented applications. *Enterprise Distributed Object Computing Workshops, International Conference on*, 0:65, 2006.

[27] Doina Caragea and Tanveer Syeda-Mahmood. Semantic api matching for automatic service composition. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 436–437, New York, NY, USA, 2004. ACM Press.

[28] Matthias Klusch, Benedikt Fries, and Katia Sycara. Automated semantic web service discovery with owls-mx. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 915–922, New York, NY, USA, 2006. ACM.

[29] Florian Rosenberg, Philipp Leitner, Anton Michlmayr, Predrag Celikovic, and Schahram Dustdar. Towards composition as a service - a quality of service driven approach. *Data Engineering, International Conference on*, 0:1733–1740, 2009.

[30] Bikash Sabata, Saurav Chatterjee, Michael Davis, Jaroslaw J. Sydir, and Thomas F. Lawrence. Taxomomy of qos specifications. *Object-Oriented Real-Time Dependable Systems, IEEE International Workshop on*, 0:100, 1997.

[31] Jeroen van de Ham. *A semantic model for complex computer networks the Network Description Language*. PhD thesis, University of Amsterdam, 2010.

[32] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. Rql: a declarative query language for rdf. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 592–603, New York, NY, USA, 2002. ACM.

[33] Abraham Bernstein and Christoph Kiefer. Imprecise rdql: towards generic retrieval in ontologies using similarity joins. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1684–1689, New York, NY, USA, 2006. ACM.

[34] Jeen Broekstra and Arjohn Kampman. Serql: A second generation rdf query language. In *Proceedings of the Workshop of SWAD-Europe Workshop on Semantic Web Storage and Retrieval*, 2003.

[35] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34:16:1–16:45, September 2009.

[36] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. In *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*, page http://www.w3.org/Submission/SWRL/, New York, NY, USA, 2004. W3C.

[37] Swi-Prolog. The homepage of swi-prolog. In *http://www.swi-prolog.org/*, 2005.

[38] Sandia National Laboratories. Jess, the rule engine for the javatm platform. In *http://www.jessrules.com/*, 2011.

[39] 2006 gridnets paper. https://oscars.es.net/oscars/docs/papers/gridnets.pdf.

[40] The opendrac network provisioning software. http://www.opendrac.org.

[41] Fenius. http://code.google.com/p/fenius/.

[42] Network service interface, http://forge.ogf.org/sf/projects/nsi-wg/.

[43] Arie Taal, Paola Grosso, Jeroen van der Ham, and Cees de Laat. Path finding strategies for multi-domain multi-layer network architectures. In *the proceedings of CGW'10*, 2010.

[44] Michele Ruta, Giammarco Zacheo, Luigi Alfredo Grieco, Tommaso Noia, Gennaro Boggia, Eufemia Tinelli, Pietro Camarda, and Eugenio Sciascio. Semantic-based resource discovery, composition and substitution in ieee 802.11 mobile ad hoc networks. *Wirel. Netw.*, 16:1223–1251, July 2010.

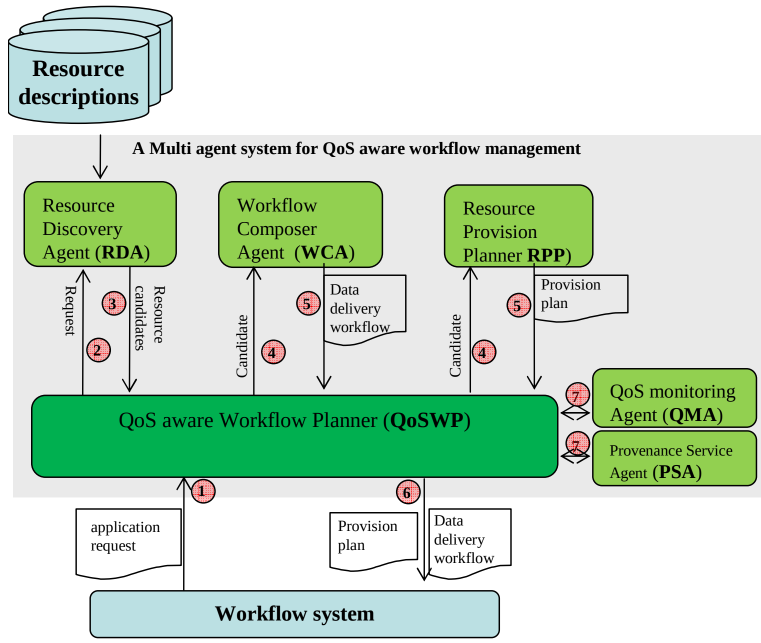

Figure 1: An agent based solution to adaptive QoS aware workflow management.

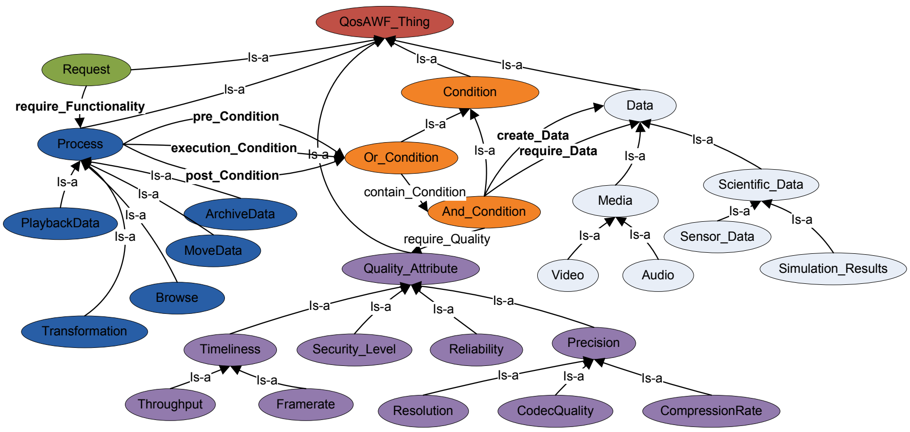

Figure 2: The concepts defined in the QoSAWF schema.

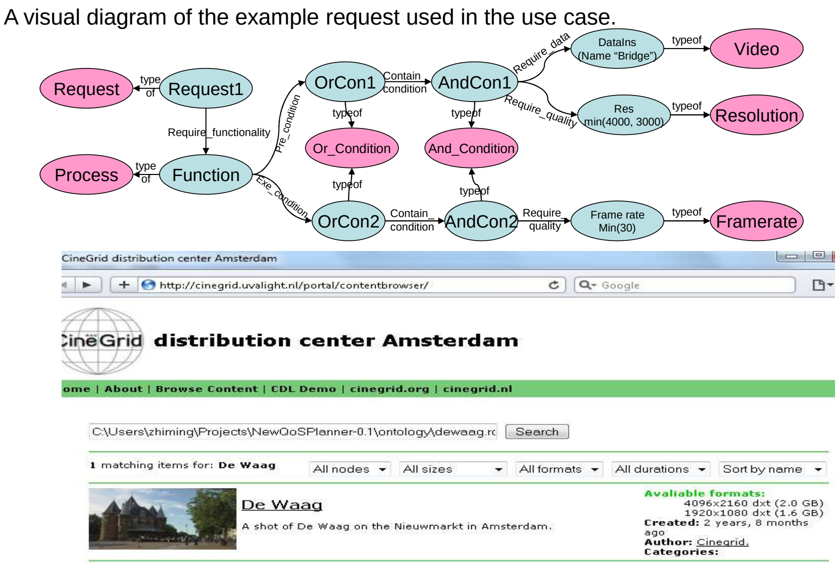

Figure 3: The top part is the visual diagram of the request, the bottom part shows a screen snapshot of the portal. The portal provides GUI components for user to describe the searching request, and can encode the user request using the QoSAWF schema and send to the planner.

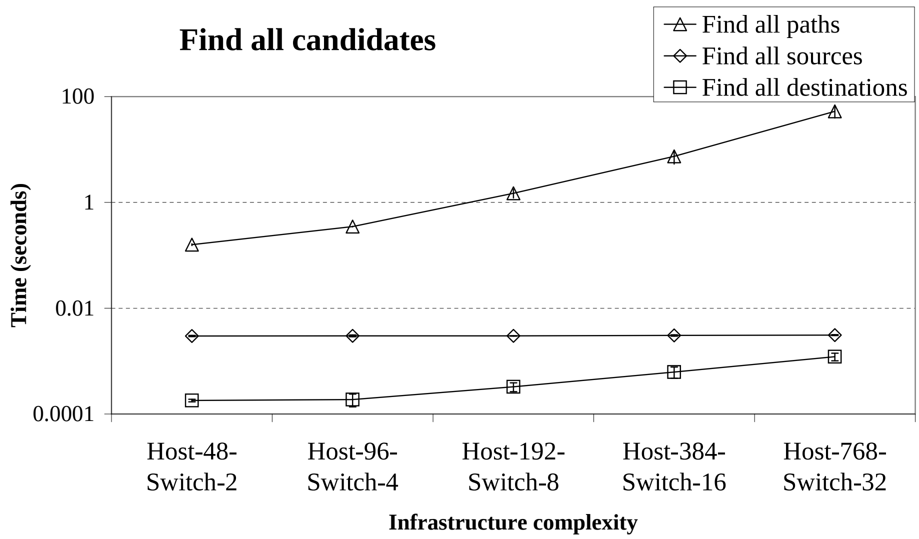

Figure 4: Time costs for finding all candidates.

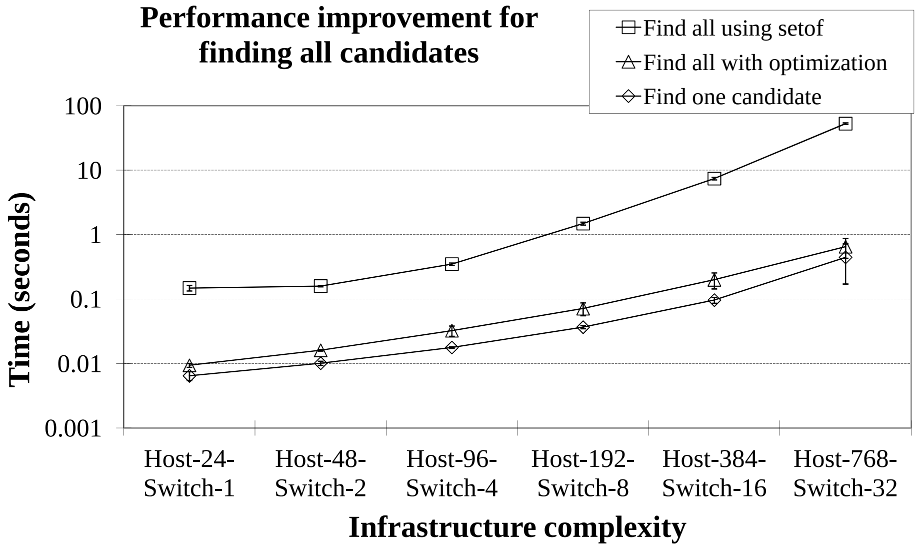

Figure 5: Optimization for finding all candidates.