EnLIGHTened Computing: An Architecture for Co-allocating Network, Compute, and other Grid Resources for High-End Applications

Lina Battestilli*, Andrei Hutanu†, Gigi Karmous-Edwards*, Daniel S. Katz†, Jon MacLaren†, Joe Mambretti‡, John H. Moore*, Seung-Jong Park†, Harry G. Perros§, Syam Sundar*, Savera Tanwir§, Steven R. Thorpe* and Yufeng Xin*
*Advanced Initiatives, MCNC, Research Triangle Park, NC, USA
{lina,gigi,jhm,sundar,thorpe,yxin}@mcnc.org
†Center for Computation & Technology, Louisiana State University, Baton Rouge, LA, USA
{ahutanu,dsk,maclaren,sjpark}@cct.lsu.edu
‡International Center for Advanced Internet Research (iCAIR), Northwestern University, Chicago, IL, USA
j-mambretti@northwestern.edu
§Department of Computer Science, North Carolina State University Raleigh, NC, USA
{hp,stanwir}@ncsu.edu

***Abstract*— Many emerging high performance applications require distributed infrastructure that is significantly more powerful and flexible than traditional Grids. Such applications require the optimization, close integration, and control of all Grid resources, including networks. The EnLIGHTened (ENL) Computing Project has designed an architectural framework that allows Grid applications to dynamically request (in-advance or on-demand) any type of Grid resource: computers, storage, instruments, and deterministic, high-bandwidth network paths, including lightpaths. Based on application requirements, the ENL middleware communicates with Grid resource managers and, when availability is verified, co-allocates all the necessary resources. ENL's Domain Network Manager controls all network resource allocations to dynamically setup and delete dedicated circuits using Generalized Multiprotocol Label Switching (GMPLS) control plane signaling. In order to make optimal brokering decisions, the ENL middleware uses near-real-time performance information about Grid resources. A prototype of this architectural framework on a national-scale testbed implementation has been used to demonstrate a small number of applications. Based on this, a set of changes for the middleware have been laid out and are being implemented.**

I. Introduction

Multiple new applications and services, especially those that are dependent on data intensive processes, require capabilities that have not been provided by traditional Grids, which usually rely on packet routed networks as non-deterministic external resources. For many of these applications, a general "best effort" service is not sufficient. Consequently, the Grid research community has been investigating new architecture and technologies that allow external networks to be used as "first class" resources within Grid environments. A number of these investigations have centered on the potential for local, regional, national, and international Grids to incorporate lightpaths, based on optical networks as key resources in Grid environments. These "optical Grids" are being investigated for their potential to support many computation-intensive scientific and commercial applications, including some that require the management of petabytes of data world-wide. To pursue these investigations experimentally, scientists in many countries have established global Grid testbeds, on which they can interconnect and share their high performance Grid facilities and use them to support real applications, not only simulations and modeled data flows [1].

While seemingly promising as a new type of communication service for Grids, optically-based distributed infrastructure also presents multiple new challenges. As a key new resource within a Grid environment, this capability must be integrated with other Grid resources and must be responsive to a changing dynamic environment and take into consideration dependencies on other resources. Accomplishing this goal is especially challenging in a dynamic environment in which the resources change while an application or workflow is being executed. Reconfigurable, dynamic optical lightpaths must be created and torn-down among multiple sites; but in coordination with the allocation, use and reconfiguration of many other Grid resources. To accomplish these goals, a new type of Grid networking middleware is necessary, along with novel transport protocols and new optical control and management interfaces. These topics have been the subject of active research in the past few years and many on-going projects are investigating them, see Appendix 2 in [2]. Some projects have been focused on allocating network resources for applications [3], [4]. However, only a few projects have focused on coordinated reservations of both network *and* other resources such as computer clusters, sensors, *etc* [5], [6].

While much progress has been made over the last decade towards developing Grid technologies, one of the key areas that has been underdeveloped is the link between Grid applications and the underlying network technologies. One of the core challenges of building Grids of supercomputers for high-end Grid applications is their interconnectivity and dealing with the large data transfers associated with executions on remote resources. Traditional Grids operate over non-deterministic, "best effort" TCP/IP networks, which do not perform well for large data transfers (terabytes or petabytes) due to the behavior of TCP over long distances. Also some Grid applications require multi-gigabits flow with low loss, low latency, and minimal jitter connecting globally distributed resources. Therefore, it is desirable to interconnect the supercomputing sites with large bandwidth, deterministic, dedicated network connections, which unfortunately can be very expensive. One way to optimize the costs is to time-share the network and only utilize bandwidth when necessary.

We are working to solve this problem by building an optical Grid, where dynamic lightpaths between the computing sites are created and torn-down based upon application needs. Grid applications are able to dynamically request in-advance or on-demand any type of Grid resource; not only high-performance computers, but also deterministic, high-bandwidth network paths. Our middleware controls all network resource allocations to dynamically setup and delete high-capacity dedicated circuits using GMPLS control plane signaling.

Currently, there is no system that can simultaneously coordinate

all resources within a Grid environment.[1] The EnLIGHTened (ENL) Computing project[2] was initiated to undertake this challenge, and also to design and develop a national experimental testbed, an optical Global Grid, with international extensions to demonstrate the effectiveness and interoperability of this architecture. Simply stated, ENL is a large-scale interdisciplinary project to address a framework for virtualizing a set of scarce resources in a coordinated manner to accomplish an objective/task(s) for a particular amount of time (in-advance or on-demand). The ENL design includes considerations of these elements:

- Application/user-initiated resource (network, compute and instruments) reservation requests
- Co-allocating various types of resources in-advance and/or on-demand
- Dynamically establishing and deleting high-capacity dedicated circuits (via control plane signaling)
- Monitoring Grid resources and retrieving information regarding: performance, availability, policy, *etc*.
- Adapting transport protocols to the demands of applications and the status of network connections.

The remainder of this paper is organized as follows. §II describes sample Grid applications that can benefit from the EnLIGHTened Computing framework. In §III, we present the initial middleware architecture. In §IV, we describe our national-footprint network testbed and supercomputing resources. §V presents the experiments done with the EnLIGHTened Computing framework to date, some results and lessons learned. In §VI we discuss plans to augment our architecture. We conclude with §VII.

II. Applications

Today's complex problems in science and engineering typically involve large-scale data, huge experiments and computer simulations, and diverse distributed collaborations of experts. Following the Optiputer paradigm [7], if we can utilize a set of computers as a single virtual machine, we will be able to solve problems of magnitudes and complexity never reached before—but this requires eliminating the network bottleneck. In this project, our driving scientific applications fall into two categories: existing applications immediately able to use high speed networks (e.g. HD video for education and the ENL/G-lambda demonstrations as described in §V-.2, visualization and access to remote data), and new applications that will be developed along with this capability.

Visualization of Remote Data and Remote Data Access. An alternative to copying large datasets to local machines for analysis is to access the data remotely from where it is stored while the visualization is running [8], [9], [10]. In this case, remote resources can perform operations on the data such as: data selection, subsambling, downsampling, caching, and executing feature extraction routines [11]. Here, the limiting factors are network bandwidth and end-to-end performance of the transport protocol.

Collaborative Visualization, Interactive Remote Visualization. Centrally-performed visualization streamed to participants [12], [13] can be used for either collaborative visualization or for interactive remove visualization, without requiring many local resources. This requires sustained high-speed transfer and low latency for the video data, and multicast capabilities. The latency for transmitting steering commands also must be minimal, as the latency of this channel plus the latency of the video channel determines interactiveness.

Traditional Distributed Computing. Distributing simulations across several high performance compute resources provides one mechanism to deal with the size and complexity of modern multi-model, multi-scale computations. There are two key classes of such applications: workflows (including parameter sweeps), and distributed applications. Workflow applications (e.g. Montage [14]), involve orchestration of compute and data transfer stages, and the mapping of these to a Grid is dependent on the availability of compute and network resources. The ability to reserve network paths will impact the time needed to complete the workflow, and is crucial for many time-critical or real-time workflows. Distributed applications (e.g. NEKTAR and Vortonics [15]), where processes of a single MPI job are distributed over multiple physically-separated Grid resources to allow larger problems to be solved. Another example of this is the G-lambda application described in [16]. Newer scenarios, where coupled models require data exchange between machines every iteration, have different needs. Here, good performance requires very high network bandwidths and very low network latencies, or a hybrid programming model that allows for the different level networks. The desired hybrid programming is not yet common in today's applications, but may become more common since it is also required for new architectures with highly-multi-core architectures or complex internal network topologies. Even the use of pure MPI applications on a Grid computing environment will allow problems to be solved that otherwise could not be. For any given machine, it is apparent that adding another machine will permit larger jobs to be run than could be run on that machine alone, with the quality of the connection between the machines affecting the speed of the larger job.

Novel Distributed Computing. Applications are also being developed that include combinations of the above types, and that are prototyping the use of high speed networks to enable entirely novel scientific investigation methods. For example, at LSU, the astrophysics code Flow-er [17] calculates stellar fluid flows on multiple computers and uses yet another computer for visualization. The Cactus framework has investigated the use of Grid paradigms for science, including task spawning, simulation steering, job migration, task farming, and large scale distributed MPI simulations [18], [19]. Other prototype applications have also demonstrated the great potential of high-speed networks as an enabler of science and engineering, and this is an area where new network capabilities will lead to new applications and modalities of scientific investigation.

III. EnLIGHTened Middleware Architecture

The current EnLIGHTened Middleware Architecture is based around, the Highly-Available Resource Co-allocator (HARC) [20], [21]. It is an open-source system that allows clients/applications to reserve multiple distributed resources in a single step. These resources can be of different types, e.g. supercomputers, dedicated network connections, storage, the use of an instrument, etc. Currently, HARC can be used to reserve both high-performance computing resources and lightpaths across certain GMPLS-based networks with simple topologies.

As shown in Fig. 1, HARC consists of a set of *Acceptors*, which manage the co-allocation process, and a set of *Resource Managers*,

[1] At time of writing both the G-lambda and EnLIGHTened projects have working systems capable of coordinating compute and network resources.

[2] EnLIGHTened Computing (http://www.enlightenedcomputing.org/) was initiated in late 2005 as an NSF seed-funded project (Award 0509465.) It is a large-scale, collaborative effort among four core partner institutions: MCNC, LSU/CCT, NCSU, and RENCI, and has grown to include several active industry participants (Cisco, Calient Networks, IBM, AT&T Research), national collaborating institutions (StarLight, Caltech, UNCC), and international collaborating projects (Japan's G-lambda, EU's PHOSPHORUS).

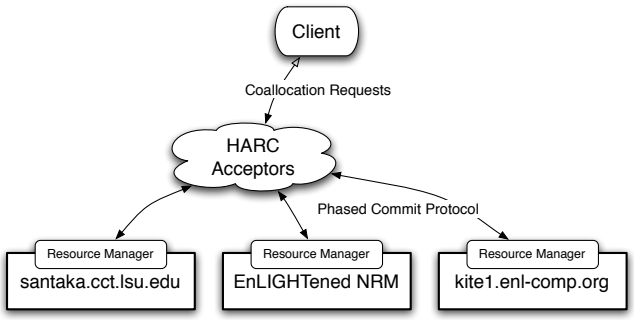

Fig. 1. Current EnLIGHTened Middleware Architecture

that provide the interfaces through which HARC makes reservations on the individual resources.

The goal of HARC is to provide a reliable service that can co-allocate resources on behalf of the user, so that the reservation process is simple, i.e. the booking process looks the same as for a single resource—a single step, with a "yes" or "no" answer. In order to book the multiple requested resources in an atomic fashion, a transaction commit protocol is required. Although 2-Phase Commit (2PC) could be used, the single Transaction Manager process would create a single point of failure, which is unsuitable in a distributed environment, where resources can fail or become unreachable. Instead, HARC uses the *Paxos Commit* protocol [22], which provides the Transaction Manager functionality through a set of cooperating processes (the Acceptors.) Due to the properties of Paxos Commit, HARC functions normally, provided a majority of the Acceptors remain operational. Clients can can communicate with any of the Acceptors at any point; consistency between the Acceptors' responses is guaranteed.

All messages in HARC are sent as XML over HTTPS, with X.509 Certificates being used to establish the SSL connections. Plain X.509 Certificates are used for all connections between the Acceptors and the RMs. GSI Proxy certificates may be used by clients when talking to Acceptors [23]. Current HARC Resource Managers authorize users using a "grid-mapfile", a simple mapping between X.509 Distinguished Names and local user names.[3]

The current implementation of HARC includes Compute Resource Managers (CRMs) and an Network Resource Manager (NRM).

A. Compute Resource Managers (CRM)

The CRMs wrap existing batch schedulers (such as BSPro, Torque with Maui, Torque with Moab, *etc*) so that HARC can make reservations on compute resources. The only requirement on a scheduler for it to be possible to use it in a HARC CRM is that it supports user-settable advance reservations.[4]

Once users have made their reservations using HARC, they submit their jobs to the reservations using the Globus Toolkit [24]. Currently, the pre-Web Services GRAM service is used for this [25], although this requires customization to allow the submission of jobs to an advance reservation rather than a batch queue.

[3]The HARC Resource Manager code could be customized to use some other form of authorization control, e.g. by calling some external Community Authorization Server.

[4]At the time of writing, all the widely used batch schedulers support this, with the exception of Sun Grid Engine; although this is planned for inclusion into SGE 6.2.

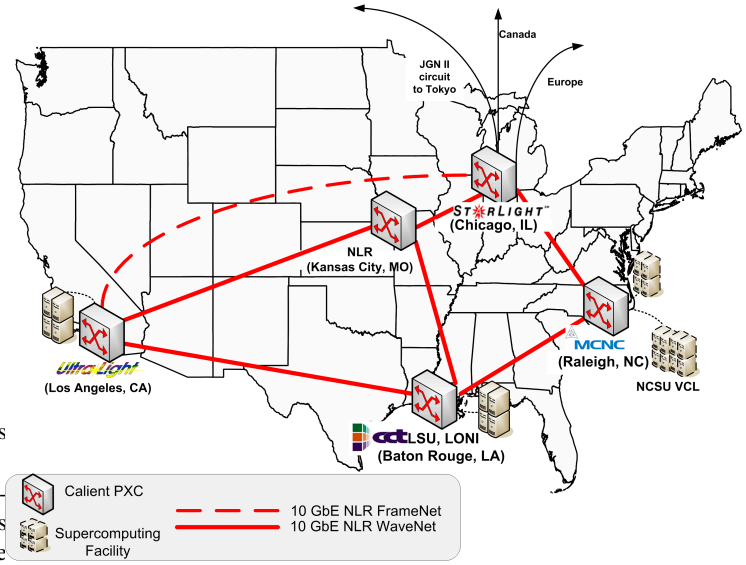

Fig. 2. EnLIGHTened Computing Testbed

B. Network Resource Manager (NRM)

How best to reserve network connectivity in advance is still a research topic; network devices do not support advance reservation of paths. When the EnLIGHTened Computing project started, there were deployed reservation systems such as the G-lambda project's GNS-WSI2 [26] and EGEE's BAR [27]. However, the project chose to implement a new, simple, timetable-based system, which was embedded in a HARC Resource Manager; this component is referred to as the HARC Network Resource Manager (NRM). There is a single, centralized HARC NRM for the entire testbed.

As stated in §IV, connections across the testbed are controlled via Calient Diamondwave switches in Los Angeles, Baton Rouge, Raleigh and Chicago. To set up and tear down a connection, the NRM sends a TL1 command to a switch at one end of the connection in order to instantiate a GMPLS command that sets up or tears down a lightpath.

IV. EnLIGHTened Resource Testbed

The ENL testbed, as shown in Fig. 2, is a national-footprint optical (Layer 1) network that has been deployed to facilitate the middleware and application goals as well as to support investigations into new network and control plane architectural models. The core of the testbed is built using Calient Networks Diamond Wave photonic cross-connect (PXC) switches interconnected by Ten Gigabit Ethernet (10 GbE) circuits provided by Cisco Systems and National Lambda Rail (NLR). GMPLS is used as the control plane protocol to allow dynamic instantiation of end-to-end paths across the testbed. Each site that hosts a PXC also provides 10-GbE-attached switches and/or end hosts as well as the potential to extend connectivity to users and resources via local Regional Optical Networks (RONs). Today, the ENL testbed interconnects five geographically-distributed supercomputing facilities: (1) Los Angeles (Caltech), (2) Baton Rouge (CCT at LSU), (3) RTP (MCNC), (4) Raleigh (VCL at NCSU) and (5) Chicago (StarLight). We also have connectivity to the JGNII testbed in Japan via a 10 Gbps circuit between Chicago and Tokyo.

V. Initial Experiments, Results, Lessons Learned

There have been three example applications demonstrated using the ENL testbed and middleware. Two application experiments have been done in cooperation with with the G-lambda project,i.e, a distributed visualization and a distributed simulation. We have also used the ENL testbed for an HD video experiment. In addition to the experiments we have done some analytical modeling and performed simulation on advance scheduling of network resources. The lessons we learned from this have led us to plan changes to the middleware, as described in §VI.

1) International Application Demonstrations: Experiments conducted between G-lambda project [5] in Japan and ENL at GLIF 2006 and SC06 demonstrated simultaneous, in-advance reservations of network lightpaths and computing resources from both the ENL and the G-Lambda testbeds. Two applications were run, one a coupled molecular dynamic and quantum mechanics application [16], and the other a distributed interactive visualization application [13]. In this collaboration, the various Resource Managers had *different interfaces* and were independently developed by the ENL and G-lambda teams. The ENL middleware used HARC to co-allocate the ENL resources. These demonstrations were successful; however, the communication between the two sets of middleware was achieved through the complicated use of software wrappers. This experience reinforced our belief that using standard, community-approved interfaces is the best way to interconnect different Grid testbeds, and has led to our current collaborations to do just this.

In addition, we learned the importance of near real-time monitoring and lightpath verification. We monitored the ongoing reservations by talking directly to the network elements and clusters. For this, we used Caltech's MonALISA [28]. We monitored our clusters, Ethernet switches and Optical switches. Using MonALISA's Optical Module we communicated to the optical switches and retrieved port optical power level and cross-connect information. We used MonALISA's GUI to watch in real-time the setting up and tearing down of the lightpaths and the load on the Ethernet switches and the clusters.

For the collaboration with G-lambda, all the Layer-2 (Ethernet) configurations were all static. All the dynamic control of the network was at done at Layer-1, i.e, the creation of lightpaths. Therefore, in the ENL testbed all the link allocations were for 10Gbps even though in some cases the application only required 1 Gbps. We realized that dynamic Layer-2 capability in the ENL testbed is a necessity and today that is a major future effort for our team.

Following the successful collaboration with G-lambda, we are now also working with the EC-funded PHOSPHORUS project [6], expanding our collaborative efforts. PHOSPHORUS is going to interconnect a number of EU NRENS, national testbeds and GLIF, in order to enable on-demand, end-to-end resource provisioning across different network domains. The EnLIGHTened, G-Lambda, and PHOSPHORUS teams met in January 2007 to discuss ways to collaborate across the three continents. We are all in strong agreement that our efforts should begin with defining standard interfaces, later discussed in §VI-.7. We are actively promoting this activity with our participation in international organizations such as the GLIF and OGF.

2) HD Video Experiment for Education: We have also used the ENL testbed to support high performance digital media streams in a distributed learning environment. In Spring 2007, LSU hosted a new high performance computing class, taught not only to students in the main LSU classroom, but also to other sites with students viewing the class over high definition or AccessGrid video, and with all students able to fully interact with the lecturer, Prof. Thomas Sterling. Additional classrooms were at LSU; LATech in Ruston, LA; UA in Fayetteville, AR; Masaryk University (MU) in Brno, Czech Republic, and NCSU/MCNC in Raleigh, NC.

Four of these sites (the second classroom at LSU, UA, MU and MCNC) were connected to the optical network. All of them were receiving a copy of the uncompressed HD video stream (1.5 Gbps bandwidth) of the lecturer sent from LSU and two of them (UA and MU) were able to send HD video streams back to the lecturer. Audio data was transmitted over the optical network between the four sites where this was available. All other communication took place using Access Grid over regular Internet connections. HARC was used to provision the connections which passed through the ENL testbed.

Uncompressed HD video is very vulnerable to packet jitter (one could use buffering but this increases delay and is undesirable in a collaborative environment) and packet loss (redundancy could be an option but if the loss is from congestion, this will make things worse). Dedicated links are currently the only infrastructure that can support this application. Since the ENL testbed is currently used for many researchers on a daily basis, it was therefore critical to utilize the ENL middleware to dynamically setup the lightpaths required by the HD-class for a short duration of time (three hours twice a week) and then release the lightpaths.

The results of this experiment demonstrated that the ENL architecture provides the capabilities required by this application. There were no problems at all with jitter or delay even with the high bandwidth demands of the application. This experiment also proves the necessity of constant network monitoring and redundant network resources. As with the collaborations with G-lambda, once again we saw the necessity for dynamic Layer-2 capability in the ENL testbed.

3) Analytical Modeling and Simulation of Advance Network Scheduling: The in-advance reservations problem in networks is a topic of current interest and it has been studied by several authors [29], [30], [31]. For our design, we chose flexible in-advance reservations, i.e. reservation requests with a scheduling window greater than the connection duration [32], as this significantly improves the network utilization. We have analytically modeled the network and carried out extensive simulation experiments to determine the best scheduling policy that minimizes the connection request blocking probability and maximizes the network utilization [33]. Our results show that minimum cost adaptive routing where link costs are determined by the current usage of the link, is the best path computation scheme. Moreover, searching for k alternate paths within the scheduling window (where k is based on the network topology) significantly improves the performance. For wavelength assignment, we chose to use a scheme that reduces fragmentation in the wavelength usage by minimizing unused trailing gaps.

We also simulated network failures. In our simulation, as soon as a link failure is detected, all the reservations on that link were re-scheduled for a *re-routing* interval. There are several ways to determine the length of this re-routing interval. A short re-routing interval results in a large number of terminated connections while a long interval re-routes unaffected connections. There is a trade-off between these two. We found that the best results are obtained when the re-routing interval is based on the moving average of the historical failure times and is updated continuously.

VI. Future Plans for the EnLIGHTened Middleware

Based on these experiments and simulations, we have planned changes to the ENL middleware. The new architecture, shown in Fig. 3, builds on the existing architecture, adding components to provide resource brokering (EnLIGHTened Resource Broker), resource

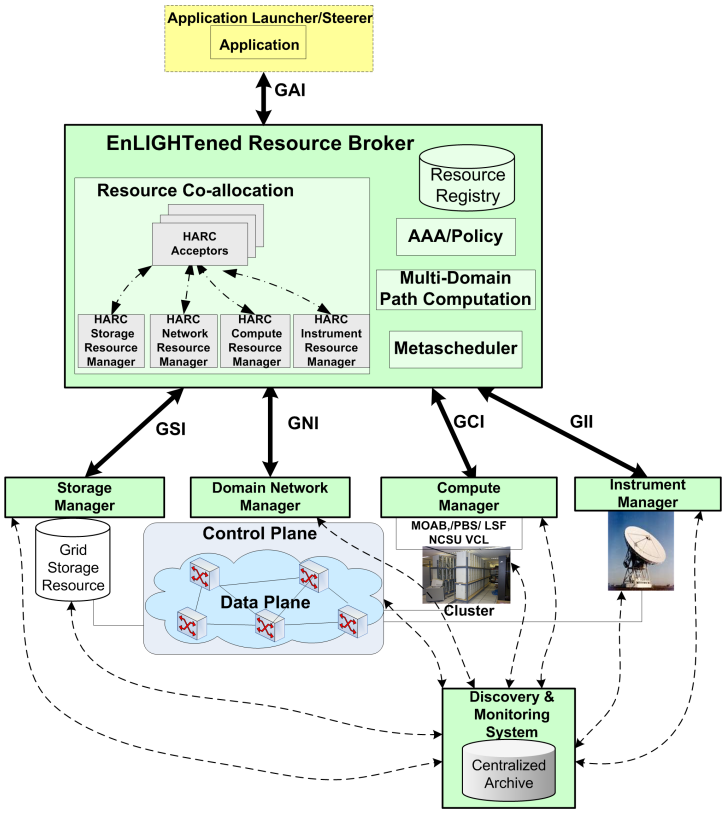

Fig. 3. Planned EnLIGHTened Middleware Architecture

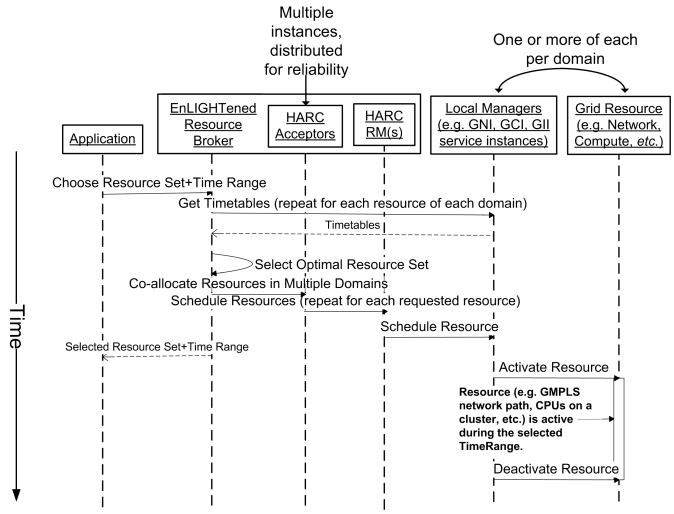

Fig. 4. Simplified diagram illustrating sequence of messages through the system when making advance reservations of resources in multiple domains

monitoring (Discovery and Monitoring System) and a much-improved network scheduling component (Domain Network Manager).

The new architecture will be used as follows. The Grid application will use the Application Launcher/Steerer (ALS) to initiate its activities. The ALS will make a reservation request for Grid resources from the EnLIGHTened Resource Broker (ERB). The ERB will query its Resource Registry and the Discovery and Monitoring System (DMS) and then choose the appropriate resources. The ERB will then talk to HARC, which will attempt to *co-allocate* the required resources for the selected time range by using the HARC Resource Managers (RM) for network, compute, storage and instruments; networking requests will be handled by the Domain Network Management component.

The rest of this section describes the new architectural components in detail.

4) EnLIGHTened Resource Broker (ERB): In the ENL middleware today, the resource and reservation time are manually selected by the user/application. We are currently working on designing the ERB, through which we will provide users with a resource brokering capability. Grid applications will send the following type of request to the ERB:

> *Select [and reserve], resource set(s) S [and time range] that can best satisfy requirements R*

where the square brackets denote optional features. The ERB will then select a resource set S (and possibly a time range) for the specific request R. The resource selection will depend upon the specified requirements, including static requirements, such as the need for a specific application or environment, and dynamic requirements, such as the deadline for the job. Resource selection will consist of the following steps.

1) Potential resources are selected by considering the static requirements (applications, environments, etc.), using information collected from the ERB's Resource Registry and from the DMS.
2) The resource list is shortened by applying dynamic requirements (processor count, deadline, etc.); this part will be achieved by requesting possible resource reservation slots from HARC.
3) A final selection is made either by applying user preferences, or by interacting with the user.

The ERB will be able to coordinate inter-domain requests, i.e, requests that require Grid resources outside a single Administrative Domain (AD). If an application requires a resource in another AD, then a path must be reserved across multiple network domains. In this case, the ERB will conduct the inter-domain coarse path-computation and resource allocation functions. The ERB will do all that using well-defined interfaces (GNI, GCI, GII, GSI) to the different types of Resource Managers. These interfaces are described at the end of this section.

Fig. 4 shows an example sequence of messages for an application making reservations of resources across multiple ADs. In this example, the application delegates the resource selection to the ERB, which in turn checks with each of the potentially usable Resource Managers for the availability status of their managed resources. Based upon the collective future state of these resources, the ERB selects a resource set, and using HARC each of the individual resources is tentatively reserved on behalf of the application. Assuming all Resource Managers respond affirmatively, the reservations are committed and the list of chosen resources is returned by the ERB to the requesting application. When the reservation time arrives, it will be the job of the Resource Managers to make the underlying resources available to the requesting application.

Earlier Grid Computing resource brokers, i.e. before the widespread availability of advance reservation, focused on the selection of suitable resources based on estimating the shortest waiting time at different resources [34], or upon a matching of application and processor count requirements of the job [35]. More recently, brokers have been developed that use reservations to deliver some

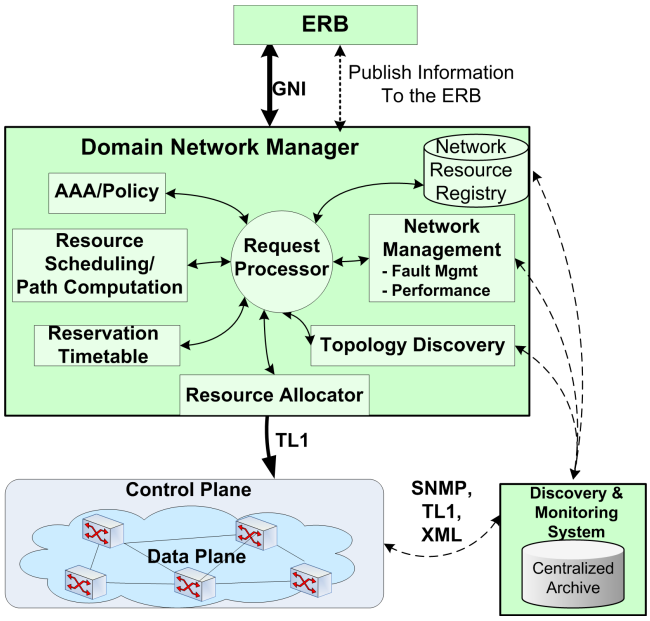

Fig. 5. Domain Network Manager Architecture

assurances about completion time, e.g. [36]. Although this earlier work will inform the ERB design, the focus of the ERB—resource brokering for multi-part jobs with compute and network co-allocation requirements—is considerably different.

5) Domain Network Manager (DNM): The new DNM will control all network resource allocations to dynamically setup and delete dedicated circuits using GMPLS control plane signaling; see Fig. 5. It will do the resource reservation (both links and wavelengths) and path computation within an Administrative Domain (AD). It will also keep the network reservation timetable. The DNM will obtain network topology information and up-to-date status of the network resources from DMS. The DNM will use its Resource Allocator to talk to the network control plane to manage (setup/teardown) lightpaths. This implies a novel relationship between the middleware and the network control plane, as the DNM will be responsible for both path computation and resource allocation. In the ENL testbed the resource allocation will be done via TL1 by instantiating a GMPLS RSVP-TE command.

The resource reservation and path computation component of the DNM will support in-advance reservations. To provide fault tolerance, the DNM will support fast restoration by using the up to date information of the network state from the DMS. For these functions of the DNM, we will use the algorithms, whose simulation results were described in §V-.3.

The DNM will store the reservation timetable for on-going and future network reservations because the network elements currently do not have a concept of schedule. It will also have a AAA/Policy component, which is very necessary in a research environment where various testbeds are interconnected and resources shared, among multiple institutions. A list of all the network resources will be stored into a registry, which will be dynamically updated by the DMS. To format network descriptions being passed among the ENL middleware components, we will utilize the Network Description Language (NDL) schemas [37] for topology, layer, capability and configuration.

6) Discovery and Monitoring System (DMS): To build a truly adaptable and resilient distributed computing environment, it is necessary to: (1) Discover and publish information about available Grid resources and use this information to make real-time resource management decisions. This is required because the resources are changing frequently: new resources and services are added, old ones are removed, capacity is increased or decreased and basic properties of a resource or service change. (2) Monitor in near real-time the availability, performance and reliability of the Grid resources for the coordination decisions of the EnLIGHTened Middleware. (3) Assess whether QoS/SLA application requirements have been met. The gathered information can be fed back to the middleware, which can dynamically renegotiate with the application to adapt to changing conditions or failures. (4) Alert Grid operators about failures, potential intrusions and warnings about conditions that may lead into failures (e.g. file system 90% full).

The EnLIGHTened DMS will have a multilayer approach to discovery, monitoring, and adaptation by using some of already existing monitoring tools [38], [39]. When available, we will also use networking equipment that comes with measurement sensors that can be queried over SNMP, XML, TL1 or by other means. This information will analyzed and used to drive decisions of the middleware.

Information about all the ENL Grid resources will stored in a centralized ENL Archive. The purpose of this archive is twofold. First, we will be able to use the collected information passively as a troubleshooting tool that will allow us to visualize the status of the resources. Second, it will provide information to the Resource Managers and the ERB. Enabling the Resource Managers and the ERB to access historical and current performance metrics of a resource will allow for better resource brokering decisions.

There are three types of network resource monitoring in the ENL project:

- *At-Reservation-Time Monitoring* refers to the monitoring steps necessary at the time a particular reservation begins. We have a reconfigurable, GMPLS-enabled Layer-1 testbed. The lightpaths between resources are automatically configured per application request when the reservation time arrives. This is very different than a Layer 3 TCP/IP network which is an always-available and sharable resource. In the ENL project, the reservation and creation of a lightpath is initiated by the middleware, which needs a mechanism to verify that indeed the network connectivity has been setup. This can be done with the following steps: (1) lightpath verification, done by communicating directly with the optical switches. (2) end-to-end IP connectivity verification, test from the application to the Grid resource. (3) Transport Protocol throughput measurement. The outputs of these three steps are communicated back to the DNM, ERB and the Grid Operators in case of failures.
- *During-Reservation Monitoring* refers to the end-to-end monitoring of an on-going reservation. It is necessary in order to assess whether the QoS/SLA application requirements have been met.
- *Ongoing Grid Resource Status Monitoring* refers to the constant, near-real time monitoring and collection of relevant performance metrics from the network resources. This information will be stored in the ENL Archive. We will follow the perfSONAR framework [40], which defines a set of services (measurement, archive, lookup, authentication, topology) and the protocol by which they communicate.

7) Interfaces Between Middleware Components: As shown in Fig. 3, all communication between the ERB and the Resource Managers is done using well defined interfaces: GNI (Grid Network

Interface), GCI (Grid Compute Interface), GSI (Grid Storage Interface) and GII (Grid Instrument Interface). The use of these interfaces will allow us to interoperate with other Grid testbeds. Currently, we are participating in discussions within GLIF's Control Plane, OGF's GHPN and NML working groups in order to define and agree upon the network-related interface, i.e, the GNI.

We have also found a need for a Grid Application Interface (GAI). Using this interface, any application can make requests to the ERB. On top of GAI, we will use the Simple API for Grid Applications (SAGA [41]). SAGA is a an OGF effort that provides a high-level programming abstraction to enable many Grid applications to be developed using simple, stable and semantically-consistent interfaces and it integrates the most common Grid programming abstractions.

VII. Conclusions

This paper has described the ongoing work in the EnLIGHTened Computing project. This project has a focus developing a framework to allow applications to virtualize a set of Grid resources (compute, instruments, sensors, dedicated network paths) in a coordinated manner to accomplish an objective/task(s) for a particular amount of time (in-advance or on-demand). Using demonstration applications, we have discovered what parts of our initial system worked well, and what changes are needed. One change is to work closely with the global community to develop standard interfaces between the various middleware implementations in order to achieve true global interoperability. Having access to the rich NLR infrastructure has provided the ENL team with a live testbed environment where we follow the research cycle: 1) develop algorithms 2) perform simulations 3) use results from simulations to develop software prototypes, 4) perform rigorous experiments on a testbed, 5) use experimental results to repeat the cycle. We also plan on continuing to refine the control of network resources and integrate novel algorithms for coordinated resource optimization.

Acknowledgments

We thank the NSF for award 0509465 which partially funded this effort, and we are grateful to many individuals without whom this effort would have been impossible: C. Hunt, A. Mabe, M. Johnson, G. Allen, L. Leger, R. Paruchuri, M. Liška, P. Holub, A. Verlo, X. Su, Y. Xia, M. Vouk, B. Hurst, D. Reed, F. Darema, K. Thompson, D. Fisher, T. West, J. Boroumand, W. Clark, R. Gyurek, K. McGrattan, P. Tompsu, S. Hunter, J. Bowers, O. Jerphagnon, M. Hiltunen, R. Schlichting, T. Kudoh, H. Nakada, A. Takefusa, Y. Tanaka, F. Okazaki, S. Sekiguchi, H. Takemiya, M. Matsuda, S. Yanagita, K. Okubo, S. Okamoto, T. Otani, Y. Sameshima, M. Suzuki, H. Tanaka, T. Otani, M. Tsurusawa, M. Hayashi, T. Miyamoto, A. Hirano, Y. SameshR. Suitte, T. Leighton, S. Rockriver, W. Imajuku, T. Ohar a, Y. Tsukishima, A. Taniguchi, M. Jinno and Y. Takigawa.

References

[1] Global Lambda Integrated Facility (GLIF). http://www.glif.is/.

[2] F. Travostino, J. Mambretti, and G. Karmous-Edwards, editors. *Grid Networks: Enabling Grids with Advanced Communication Technology*. Wiley, 2006.

[3] Dynamic resource allocation over gmpls optical networks. http://dragon.maxgigapop.net.

[4] User-controlled lightpath. http://www.canarie.ca/canet4/uclp.

[5] G-Lambda Project Website. http://www.g-lambda.net/.

[6] PHOSPHORUS Project Website. http://www.ist-phosphorus.eu/.

[7] L. L. Smarr, A. A. Chien, T. DeFanti, J. Leigh, and P. M. Papadopoulos. The OptIPuter. *Communications of the ACM*, 46(11):58–67, 2003.

[8] S. Prohaska, A. Hutanu, R. Kähler, and H.-C. Hege. Interactive Exploration of Large Remote Micro-CT Scans. In *Proc. IEEE Vis. '04*, pages 345–352, 2004.

[9] T. Kurc, Ü. Çatalyürek, C. Chang, A. Sussman, and J. Saltz. Visualization of Large Data Sets with the Active Data Repository. *IEEE Comput. Graph. Appl.*, 21(4):24–33, 2001.

[10] C. Zhang, J. Leigh, T. A. DeFanti, M. Mazzucco, and R. Grossman. TeraScope: distributed visual data mining of terascale data sets over photonic networks. *Future Gener. Comput. Syst.*, 19(6):935–943, 2003.

[11] M. D. Beynon, R. Ferreira, T. Kurc, A. Sussman, and J. Saltz. DataCutter: Middleware for Filtering Very Large Scientific Datasets on Archival Storage Systems. In *Proc. Mass Storage Systems*, pages 119–133, March 2000.

[12] L. Renambot, T. van der Schaaf, H. E. Bal, D. Germans, and H. J. W. Spoelder. Griz: experience with remote visualization over an optical grid. *Future Gener. Comput. Syst.*, 19(6):871–881, 2003.

[13] A. Hutanu, G. Allen, S. D. Beck, P. Holub, H. Kaiser, A. Kulshrestha, M. Liška, J. MacLaren, L. Matyska, R. Paruchuri, S. Prohaska, E. Seidel, B. Ullmer, and S. Venkataraman. Distributed and collaborative visualization of large data sets using high-speed networks. *Future Generation Computer Systems: The Int. J. of Grid Comp.: Theory, Methods and Applications*, 8:1004–1010, Oct. 2006.

[14] J. C. Jacob, D. S. Katz, G. B. Berriman, J. C. Good, A. C. Laity, E. Deelman, C. Kesselman, G. Singh, M.-H. Su, T. A. Prince, and R. Williams. Montage: A Grid Portal and Software Toolkit for Science-Grade Astronomical Image Mosaicking. *International Journal of Computational Science and Engineering*, 2007.

[15] B. Boghosian, P. Coveney, S. Dong, L. Finn, S. Jha, G. Karniadakis, and N. Karonis. Nektar, SPICE and Vortonics: Using Federated Grids for Large Scale Scientific Applications. In *Challenges on Large Applications in Distributed Environments (CLADE06), International Symposium on High Performance Distributed Computing Conference (HPDC-15)*, pages 32–42, 2006.

[16] S. Ogata, F. Shimojo, R. Kalia, A. Nakano, and P. Vashishta. Hybrid Quantum Mechanical/Molecular Dynamics Simulations on Parallel Computers: Density Functional Theory on Real-space Multigrids. *Comp. Phys. Comm.*, 149:30–38, 2002.

[17] P. M. Motl, J. E. Tohline, and J. Frank. Numerical Methods for the Simulation of Dynamical Mass Transfer in Binaries. *The Astrophysical Journal Supplement Series*, 138:121–148, 2002.

[18] G. Allen, D. Angulo, I. Foster, G. Lanfermann, C. Liu, T. Radke, E. Seidel, and J. Shalf. The Cactus Worm: Experiments with Dynamic Resource Discovery and Allocation in a Grid Environment. *Int. J. of High Performance Computing Applications*, 15(4), 2001. http://www.cactuscode.org/Papers/IJSA_2001.pdf.

[19] G. Allen, T. Dramlitsch, I. Foster, N. Karonis, M. Ripeanu, E. Seidel, and B. Toonen. Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus. In *Proc. of Supercomputing 2001*, 2001.

[20] HARC: Highly-Available Resource Co-allocator. http://www.cct.lsu.edu/~maclaren/HARC/.

[21] J. MacLaren. Co-allocation of Compute and Network resources using HARC. In *Proceedings of "Lighting the Blue Touchpaper for UK e-Science: Closing Conference of the ESLEA Project*. PoS(ESLEA)016, 2007. http://pos.sissa.it/archive/conferences/041/016/ESLEA_016.pdf.

[22] J. Gray and L. Lamport. Consensus on transaction commit. *ACM TODS*, 31(1):130–160, March 2006.

[23] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820, IETF, 2004.

[24] Globus Alliance web page. http://www.globus.org, February 2006.

[25] Pre-WS GRAM Documentation. http://www.globus.org/toolkit/docs/4.0/execution/prewsgram/.

[26] A. Takefusa et al. GNS-WSI2 Grid Network Service - Web Services Interface, version 2. OGF19, GHPN-RG, 2007.

[27] C. Palansuriya, M. Büchli, K. Kavoussanakis, A. Patil, C. Tziouvaras, A. Trew, A. Simpson, and R. Baxter. End-to-End Bandwidth Allocation and Reservation for Grid applications. In *Proc. of BROADNETS 2006*, October 2006.

[28] MonALISA Web Page. http://monalisa.cacr.caltech.edu, February 2006. MONitoring Agents using a Large Integrated Services Architecture.

[29] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. A Distributed Resource Management Architecture that Supports Advance Reservations and Co-allocation. In *7th IEEE/IFIP Int. Workshop on Quality of Service (IWQoS'99)*, pages 27–36, 1999.

[30] L.-O. Burchard. Networks with Advance Reservations: Applications,

Architecture, and Performance. *Journal of Network and Systems Management*, 13(4), 2005.

[31] Jun Z. and H. T. Mouftah. Routing and Wavelength Assignment for Advance Reservation in Wavelength-Routed WDM Optical Networks. In *Proceedings of the IEEE International Conference on Communications, ICC'02*, volume 5, 2002.

[32] E. He, X. Wang, and J. Leigh. A Flexible Advance Reservation Model for Multi-Domain WDM Optical Networks. In *Proc. of GridNets 2006*, San Jose, Oct. 2006.

[33] S. Tanwir, T. Battestilli, H. Perros, and G. Karmous-Edwards. Dynamic scheduling of network resources with advance reservations in optical grids. *Submitted for publication*.

[34] G. Aloisio, M. Cafaro, E. Blasi, and I. Epicoco. The Grid Resource Broker, a Ubiquitous Grid Computing Framework. *Sci. Prog.*, 10(2):113–119, 2002.

[35] John Brooke, Donal Fellows, and Jon MacLaren. Resource Brokering: The EUROGRID/GRIP Approach. In *Proceedings of the UK e-Science All Hands Meeting 2004*. http://www.allhands.org.uk/2004/proceedings/papers/178.pdf, 2004.

[36] E. Elmroth and J. Tordsson. A Grid Resource Broker Supporting Advance Reservations and Benchmark-based Resource Selection. In *Applied Parallel Computing: State of the Art in Scientific Computing*, number 3732 in Lecture Notes in Computer Science, pages 1061–1070. Springer-Verlag, 2006.

[37] Network Description Language Schema. http://www.science.uva.nl/research/sne/ndl/?c=01-NDL-Schema.

[38] S. Zanikolas and R. Sakellariou. A Taxonomy of Grid Monitoring Systems. *Future Generation Computer Systems*, 21(1):163–188, January 2005.

[39] SLAC's List of Tools Used for Network Monitoring. http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html, Feb. 2006.

[40] perfSONAR web page. http://www.perfsonar.net, January 2007.

[41] SAGA Research Group. https://forge.gridforum.org/projects/saga-rg.