

Collaborative Data Visualization for Earth Sciences with the OptIPuter

Nut Taesombut, Xinran (Ryan) Wu, and Andrew A. Chien
Department of Computer Science and Engineering, and Center for Networked Systems
University of California, San Diego, La Jolla CA 92093
{nut, xwu, achien}@cs.ucsd.edu

Atul Nayak, Bridget Smith, Debi Kilb, Thomas Im, Dane Samilo, Graham Kent, and John Orcutt
Cecil H. and Ida M. Green Institute of Geophysics and Planetary Physics
Scripps Institution of Oceanography
University of California, San Diego, La Jolla CA 92093
{anayak, brsmith, dkilb, thim, dsamilo, gkent, jorcutt}@ucsd.edu

Abstract

Collaborative visualization of large-scale datasets across geographically distributed sites is becoming increasingly important for Earth Sciences. Not only does it enhance understanding of the geological systems, but also enables near-real-time scientific data acquisition and exploration across distant locations. While such a collaborative environment is feasible with advanced optical networks and resource sharing in the form of Grid, many technical challenges remain: (1) on-demand discovery, selection and configuration of supporting end and network resources; (2) construction of applications on heterogeneous, distributed environments; and (3) use of novel exotic transport protocols to achieve high performance. To address these issues, we describe the multi-layered OptIPuter middleware technologies, including simple resource abstractions, dynamic network provisioning, and novel data transport services. In this paper, we present an evaluation of the first integrated prototype of the OptIPuter system software recently demonstrated at the iGrid2005, which successfully supports real-time collaborative visualizations of 3D multi-gigabyte earth science datasets.

1. Introduction

Collaborative visualization of large-scale datasets across geographically distributed sites is becoming increasingly important for Earth Sciences. Today, a significant amount of earth science data is being generated from remote sources such as wireless sensors or satellites. These scientific data is massive, comprising sets of objects as large as several gigabytes and collections larger than hundreds of terabytes. For example, modern 3D seismic volumes of Earth's substructures can be as large as 50GB [1]. Capitalizing on the availability of these high-resolution images, work is underway to enable multi-dimensional visualization of these objects, enhancing

the understanding of the complex geological systems such as the volcanic development and deformation of Earth's surfaces. Furthermore, remote and collaborative visualization [1] enables a group of scientists from distantly located institutions to interactively analyze the collected data in real-time, thereby increasing the productivity of scientific data interpretation.

The ability to build a wide-area, collaborative visualization environment is made possible by continuing advances in network capabilities enabled by Dense Wavelength Division Multiplexing (DWDM) [2], and cross-domain resource aggregation in the form of Grid [3]. DWDM is an efficient technique that enables a single fiber to carry multiple wavelengths (or lambdas), increasing an aggregate throughput on each fiber as high as several terabits per second. These private, high-speed optical paths can be dynamically configured to interconnect remote storage, computation and visualization resources across wide-area networks in seconds. Collaborative visualization applications benefit from these private networks because they provide secure, ultra-high-speed congestion-free channels, which guarantee network performance such as bounded jitter and delay. Exploiting this trend, an increasing number of lambda network testbeds have been deployed, including OptIPuter [4], National Lambda Rail, Dragon, CHEETAH, Global Lambda Interchange Facility (GLIF), CANARIE's CA*net 4, and Netherlight.

While the hardware requirements of collaborative visualization environments can be met by the current and emerging infrastructures, building these applications is difficult due to many reasons:

- Identifying and selecting end and network resources in the system requires understanding of the complex software and hardware infrastructures,

- Utilizing resources in wide-area networks involves management of cross-organization security, heterogeneous resource capabilities and system failures,
- Employing configurable networks requires management of multi-domain routing, signaling and dynamic resource naming, and
- Achieving high and robust network performance involves the use of novel exotic transport protocols

In this paper, we present the OptIPuter middleware, a multi-layered integrated solution for building distributed applications on Lambda-Grids [4]. The middleware allows applications to dynamically configure end and private network resources for their simple and robust execution. It addresses the above challenges by integrating novel capabilities of Grid and network services, and presenting a unified simple resource abstraction to applications. At the iGrid2005 workshop (www.igrid2005.org), we made the first-time demonstration of the performance of the integrated OptIPuter middleware, and its capabilities of supporting real-time collaborative visualization of 3D multi-gigabyte earth science datasets. In this paper, we present the evaluation and demonstration results. Specific contributions of this paper include:

- a description of the first-time integration of multi-layered OptIPuter software technologies, including novel data transports, simple resource abstractions, and dynamic network provisioning,
- an evaluation of the first-time demonstration of the OptIPuter middleware with a real scientific application, highlighting the enabled capabilities and simple application construction, and
- an evaluation of the developed middleware prototype on the OptIPuter's international-scale Lambda-Grid testbed, including the performance of high-speed data transfer and resource configuration

We organize this paper as follows: In Section 2, we describe collaborative visualization applications for Earth Sciences. In Section 3, we present the architecture of the OptIPuter middleware and its components. In Section 4, we show the performance of the OptIPuter middleware prototype, and its capabilities for enabling a collaborative visualization environment. We conclude the paper in Section 5.

2. Collaborative Data Visualization for Earth Sciences

Various fields in the geosciences are seeing a dramatic rise in the volume and quality of data being collected from regional and global-scale observing systems or generated by simulation of theoretical

models. Advanced visualization tools can be applied to enable scientists to interactively explore visualized data objects at very high resolution and in multiple dimensions, enhancing understanding of complex geological systems.

For example, researchers at the Scripps Visualization Center are using the datasets from EarthScope to study the activity of the San Andreas Fault in California. They use a visualization package called 'Fledermaus' (www.ivs3d.com) which imports raw geophysical datasets and arranges them in a geo-referenced coordinate system. The visualization can be saved as a 'scene' file representing the 3D strain fields resulting from deformation of the Earth's crust.

This technique of data visualization has become very popular within the Scripps Institution and its collaborating agencies. To share this data and visualizations with their collaborators, the Scripps researchers create the scene files and make them available for download. However, one major challenge blocking this scientific data sharing is that individual files are large and a vast number of them are often required for data correlation analysis. Until recently, because of limited capacity of the traditional shared Internet, such sharing is restricted to small scene files or within local-area environments.

The OptIPuter's objective is to enable wide-area scientific collaborations [1], where scientists can interactively analyze and collaboratively visualize very large data objects at very high resolution and with other distantly located scientists. Such collaborations support the ability to transfer multi-gigabyte scene files between collaborating sites in real-time and on-demand, and to visualize them on ultra-high-resolution display systems such as 'LambdaVision' [5]. Parallel visualizations of these datasets on multi-tiled display LambdaVision allow scientists to perform time-series analysis of complex systems such as that of earthquake activities over the past century.

3. The OptIPuter Middleware

The OptIPuter middleware is an integrated set of novel software technologies which enable development of high-performance, distributed applications on Lambda-Grids. It provides a simple way for applications to acquire and use communication and end resources (compute, storage, visualization), while ensuring their robust execution.

Figure 1 shows the layered software architecture of the OptIPuter middleware. To simplify use and management of distributed

resources for applications, our approach is to provide a resource abstraction, the Distributed Virtual Computer (DVC), which encapsulates the complexity of underlying software and hardware infrastructures, while presenting applications a simplified interface. The DVC software layer integrates emerging Grid, dynamic optical network provisioning, novel transport and distributed storage services, and leverages these capabilities to applications.

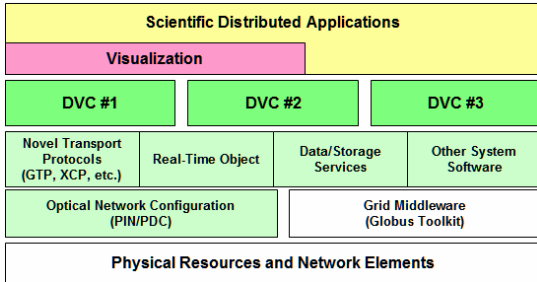


Figure 1. OptIPuter software architecture (green boxes indicate OptIPuter software technologies)

The following subsections describe three key components of the OptIPuter middleware currently implemented and integrated to support Earth Sciences' collaborative visualization environments.

- *Distributed Virtual Computer (DVC)* is a set of high-level services which enable simple resource abstractions for easy and efficient application construction on Lambda-Grids.
- *Photonic Inter-domain Negotiator (PIN) and Photonic Domain Controller (PDC)* is an implementation of the multi-domain dynamic lambda provisioning services.
- *Group Transport Protocol (GTP)* is an end-node based data transport protocol which features efficient and fair sharing of end-node capacity across active connections.

In fact, the OptIPuter middleware effort spans a broader range of system research areas, including distributed storage and file systems, multi-point communication abstractions, visualization, and real-time object models. The details of these components are beyond the scope of this article. Interested readers should refer to [1,6,7].

3.1 Distributed Virtual Computer (DVC)

A DVC [8] is a middleware-enabled, configurable collection of private Grid and communication resources usable by applications to achieve simple, reliable and high-performance execution. It provides a powerful set of abstractions that shield applications from the complexities of underlying infrastructures and presents them a simple

view of tightly-coupled local clusters instead of a complex distributed environment. Within a DVC, applications can have direct, secure, reliable and high-performance access to remote resources (i.e., SAN-like use across geographically distributed sites). As shown in Figure 2, the DVC abstractions are realized by a set of cooperating services and simple interfaces for resource management and communication.

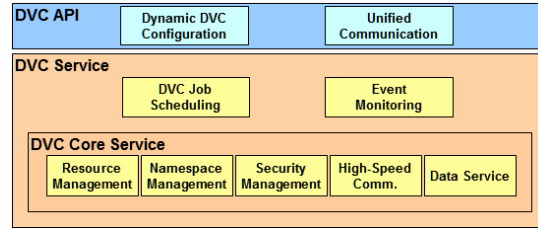


Figure 2. DVC service architecture

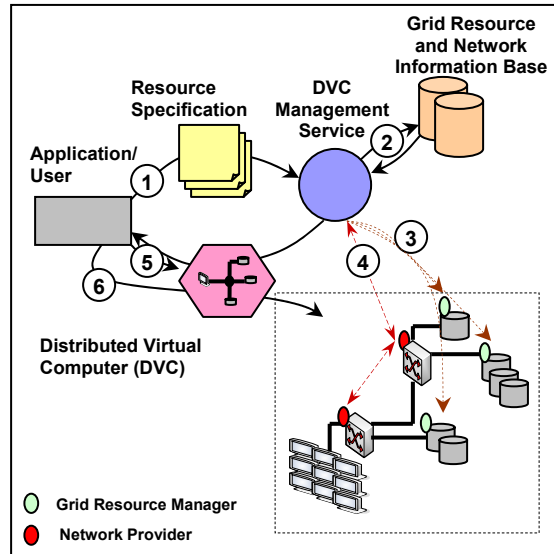


Figure 3. DVC service model: (1) description of resource needs; (2) coordinated resource discovery and selection; (3) Grid resource allocation; (4) dynamic network setup; (5) configuration of a virtual namespace and communication; and (6) application execution

DVC's allow applications to describe, acquire and configure a set of distributed resources and private networks for their simple and high-performance execution. As shown in Figure 3, to create a DVC, an application specifies its end and communication resource needs and presents them to the DVC resource management service. Subsequently, the service identifies the resource candidates and computes a physical resource configuration (end resources + networks) that matches all the requirements. If satisfied, the chosen configuration is realized transparently by a set of cooperating services that negotiate with Grid

resource managers and network providers. The allocated resources are bound into a newly created DVC with configurable trust relationships, communication, resource namespace, and performance control.

3.1.1 Integrated, Declarative Expression of Resource Needs

The DVC model enables applications to express their end and communication resource needs in one integrated language, called *DVC Integrated Specification Language (DVC-ISL)*. The language builds on the RedLine constraint language [9] and extends it with simple abstractions for expressing novel communication structures such as private networks with shared internal connection and photonic multicast [10]. A key novelty of the language is the explicit expression of communication resources, which allows users to optimize network configuration for their applications (either for efficient resource use or high application performance). For example, a desired routing path for communication between two end resources can be explicitly specified by a chain of OXCs' addresses.

```
(1): viz-cluster ISA SET [InSet(SpecialHW, "11x5 tiled-display"); Count(viz-cluster) == 25;
(2): str-cluster1 ISA SET [InSet(DataSet, "SoCalSAFS00-40"); Count(str-cluster1) == 13;
(3): str-cluster2 ISA SET [InSet(DataSet, "SoCalSAFS41-80"); Count(str-cluster2) == 7;
(4): str-cluster3 ISA SET [InSet(DataSet, "SoCalSAFS81-99"); Count(str-cluster3) == 5;
(5): conn1 ISA CONN (<viz-cluster>,<str-cluster1>) [type="Lambda"; Bandwidth >= 10000];
(6): conn2 ISA CONN (<viz-cluster>,<str-cluster2>) [type="Lambda"; Bandwidth >= 10000];
(7): conn3 ISA CONN (<viz-cluster>,<str-cluster3>) [type="Lambda"; Bandwidth >= 4000]
```

Figure 4. A sample DVC-ISL specification for a collaborative visualization application

A sample DVC-ISL specification for a collaborative visualization application appears in Figure 4. The “ISA SET” and “ISA CONN” operators are used to describe the need for end resource group/set (e.g., computing, storage and visualization clusters) and communication resource (e.g., network connection), respectively. The sample specification requests for one visualization cluster (line 1), three storage clusters (line2-4) and lambda connectivity between them (line5-7). The visualization cluster must be connected to a 55-panel tiled display and have 25 rendering nodes. The storage clusters must store the specified datasets “SoCalSAFS*” and have certain numbers of nodes. Further, it requires lambda connectivity (10Gbps and 4Gbps) between the visualization cluster and individual storage clusters.

The example specification illustrates an advantage of the DVC-ISL language to specify the group connectivity between visualization and storage machines across clusters (*N-to-M* cross-cluster

communication). It allows collections of machines to share physical network connection, resulting in efficient utilization of network resources.

3.1.2 Integrated Resource Management

To simplify and optimize application use of Lambda-Grid resources, DVC’s provide an integrated resource management service for optimized resource planning, allocation and performance control. Applications describe their resource needs, and the service transparently discovers, selects and realizes the matching configurations which combine distributed resources with a set of dynamically configured optical networks. This automated process shields the applications from the complexities of low-level dynamic network configuration and Grid resource acquisition, thereby simplifying their construction.

By combining the acquisition of communication, computing, and storage resources, the entire resource needs of an application can be effectively addressed, and supported. Such combined acquisition enables coordinated resource selection which can improve resource efficiencies as well as better application capabilities. For example, when the interested datasets are replicated on multiple remote repositories, the system can identify and allocate one with the best connectivity to visualization. (Separate selection may result in poorer resource use efficiency and application performance due to limited degree of choices).

To support robust application execution, the DVC resource management service provides monitoring and event subscribe/notify systems, which allow applications to be notified and react promptly to changes in resource conditions. In the future, our software will incorporate dynamic reconfiguration capabilities which allow DVC’s to transparently address resource failures and optimize application execution.

3.1.3 Configurable, Virtual Resource Names

To enable applications to run on different physical resource configurations without modification, DVC’s provide a private virtual namespace. Within a DVC, each resource is assigned a unique virtual IP address and hostname. An application may define these names and use them to avoid dealing with their heterogeneous physical addresses (e.g., dynamic IP addresses) or for other convenient use. When these virtual addresses are used for communication, DVC’s map them to the corresponding physical resource addresses and redirect traffic to the appropriate

destinations. Because the same virtual namespace can be assigned to different resource sets, the application can be independently run on them.

3.1.4 Unified Communication APIs

To simplify application use of novel protocols [11-16], DVC's provide an integrated communication framework that combines different protocol implementations and provides a simple, unified set of interfaces to the applications. These protocols are essential to achieve high performance on long-haul networks, though having implicit complexities and presenting diverse interfaces. The DVC communication APIs include familiar pair-wise and collective communication interfaces defined against the virtual resource names (See Section 3.1.3). The unified APIs allow the applications to transparently utilize different communication protocols and/or mechanisms (application-enabled multicast vs. photonic multicast), enabling them to easily adapt to diverse resource conditions.

3.2 Dynamic Network Provisioning (PIN/PDC)

Photonic Inter-domain Negotiator (PIN) [17, 18] is a distributed agent architecture for dynamic lightpath provisioning on optical networks across domains. It addresses the complex issues of inter-domain routing, signaling and security for optimal lightpath scheduling and configuration.

The architecture consists of distributed PIN agents in heterogeneous network domains which may employ diverse schemes for routing, signaling, and security. To establish a lightpath connection between two end hosts, an application issues a request to a local PIN agent which propagates the request message to remote domains until the destination is reached. For each domain along the path, the corresponding PIN agent translates the request into a native signaling message understood by the local network control facility. The facility then configures its local switches to establish a new connection. PIN relies on AAA [19] for secure network configuration.

PIN is a flexible framework which supports a broad range of intra-domain network control planes. The current implementation relies on Photonic Domain Controller (PDC) [20] which can provision end-to-end lightpaths within a single domain. PDC comprises modules for intra-domain routing and interfacing with local MEMS switches for dynamic lightpath setup and teardown.

3.3 Group Transport Protocol (GTP)

DWDM enables plentiful bandwidth (several 10's Gbps) of lambda networks, pushing resource contention and sharing bottlenecks to end systems

(i.e., slow disk or processing speed to absorb the bandwidth). Application trends have been increasing the use of multipoint-to-point traffic patterns (as evidenced in the world-wide web, P2P networks, and large-scale scientific data sharing). High-speed core networks enable many fast flows to converge on an endpoint, raising challenges in efficient and fair sharing of source and sink capacities.

The Group Transport Protocol (GTP) [11] targets efficient and fair sharing of source and sink node capacities among active connections in such high-speed long-distance network environments. GTP features rate-based explicit flow control and end-node based max-min fair rate allocation across multiple flows at the same end-node to support multipoint-to-point and multipoint-to-multipoint data movement. GTP's novel distributed rate allocation algorithm controls each source and sink independently with local information and adaptively allocating its capacity to candidate sessions. This approach enables the bandwidth of multiple flows to converge rapidly to max-min fair rate allocation and is in practice fast in networks of 64 to 1024 nodes [21].

Our other studies [22] show that GTP performs as well as other UDP based aggressive transport protocols (e.g. RBUDP[16], SABUL[14]) for single flows, and when converging flows (from multiple senders to one or multiple receivers) are introduced, GTP achieves both high throughput and much lower loss rates than others.

During iGrid2005, we demonstrated the performance of GTP by moving large earth science datasets concurrently from multiple remote data repositories (Amsterdam, Chicago, and UCSD) to the iGrid2005 visualization cluster.

4. Experimental Studies

To evaluate the performance of the OptIPuter middleware and to demonstrate its capabilities in supporting remote and collaborative visualization environments, we conducted experiments using the integrated OptIPuter middleware technologies (including GTP, PIN/PDC and DVC) at the iGrid2005. The collaborative environment features 5-layer software demonstration (See Figure 1), including scientific collaboration for earth science, interactive 3D visualization (Fledermaus), DVC resource abstractions, high-speed data transfer (GTP), and dynamic network allocation (PIN/PDC). This is the first-time demonstration of integrated, end-to-end OptIPuter software technologies with a real scientific application.

In the following subsections, we describe the experiment configurations and stage-by-stage results.

4.1 Demonstration Setup

In the demonstration, we used the OptIPuter’s international Lambda-Grid testbed [23] and the network infrastructure provided by iGrid2005. As shown in Figure 5, the infrastructure consists of distributed storage clusters across sites at UCSD/San Diego (5 sites on campus), UIC/Chicago and UvA/Amsterdam, and the visualization system (55-tiled display wall driven by the 28-node cluster) in the Calit2 building at UCSD. The visualization system has two 10 Gbps uplink interfaces with aggregate 10 Gbps connectivity to/from the other sites at UCSD and another aggregate 10 Gbps connectivity to/from UvA and UIC. The UCSD OptIPuter network is controlled by a configurable MEMS-based optical cross-connect (OXC) switch, which is capable of dynamically switching connection from one site on campus to the visualization system.

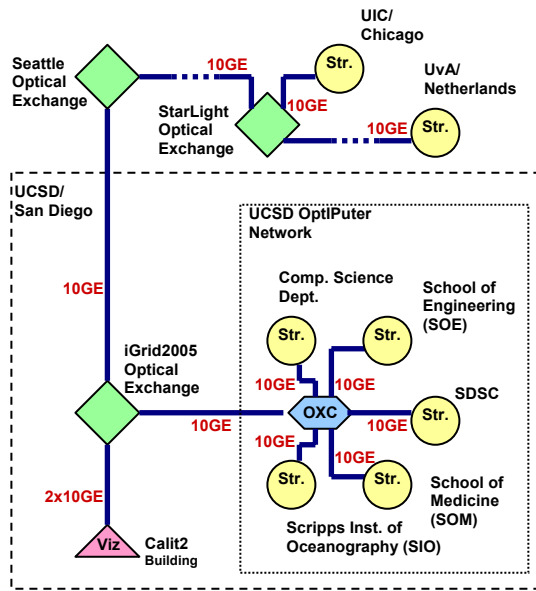


Figure 5. OptIPuter testbed and iGrid2005 networking infrastructure

4.2 Demonstration Results

To establish the required collaborative visualization environment, we first created a DVC, a virtual resource environment that consists of three storage clusters, one visualization cluster (with the 100 mega-pixel LambdaVision tiled display at Calit2), and lambda connectivity between them. We then created a virtual namespace, assigning individual end resources with unique logical names, and built an application that utilizes them. We

configured the DVC to use GTP as the default transport protocol. We launched an interactive application that allows the user to choose interested datasets, transfers them from remote storages in real-time using GTP, and then uses Fledermaus, an interactive 3D visualization system, to display all the requested data simultaneously on the 55-tiled display panel (we actually used 50 out of 55 displays).

4.2.1 On-demand Resource Selection and Allocation

The first step in enabling a collaborative environment is on-demand resource discovery, selection and allocation. We submitted a resource specification (See Figure 4) to the DVC resource management service. The service replied that a resource configuration that consists of storage clusters at SOE/UCSD, UIC, and UvA and a visualization cluster at Calit2 was available. These three storage clusters were picked because the entire library of datasets existed on the disks. Also when we conducted this experiment at iGrid 2005 only one storage cluster at UCSD could be connected to the visualization cluster because of limitation in the networking infrastructure at UCSD (on other runs of the experiment, one other storage cluster at UCSD can be selected). The chosen configuration also includes the network configuration which included the dynamic lightpath setup between the SOE cluster and the visualization cluster. The connection between the visualization cluster and the clusters at UvA and UIC was static and already in place; no dynamic network setup is required.

The DVC automatically realized the chosen configuration, including dynamic network setup and end resource allocation. The end resource allocation is done through the GRAM service of the Globus Toolkit 3.2, while the network configuration is done using PIN/PDC. In all, we allocated 50 cluster machines and dynamically modified one OXC switch configuration. We measured the time since the query for resources upto the completion of the configuration. Table 1 breaks down the time in each step.

Table 1. Resource Configuration Performance

Resource Selection	End Resource Allocation	Dynamic Network Configuration	Total Time
0.754s	3.455s	1.238s	5.447s

The total resource configuration time is 5.447 seconds, where 13.84%, 63.43% and 22.73% of the time is spent on resource selection, grid resource allocation and dynamic network setup, respectively.

The majority of resource selection time involves the cost of resource information query on a local database, resource matching and network configuration planning. The selection cost depends on the complexity of application request and is likely to grow with the number and quality of required resources. Both resource matching and network planning are NP-hard problems [24,25] and the current subject of our research. At iGrid2005, we implemented the DVC resource selection service with meta-heuristic algorithms based on Simulated Annealing [26].

The end resource allocation cost is dominated by the use of GRAM to start up DVC control daemons on 50 remote resources. This can be done using multiple calls to GRAM in either blocking or non-blocking modes. We performed two runs using different modes and observed significant reduction in allocation time when using non-blocking calls (3.455s) compared to when using blocking calls (68.291s). The use of non-blocking calls permits parallel resource allocation which reduces the impact of the long waiting time between calls as a direct result from remote process startup.

The network setup cost is attributed to the instantiation and operation of PIN/PDC web services (based on OGS/OGSA). The operation includes network information query on a local database, translation of a request into a physical configuration plan and reconfiguration of an OXC switch.

4.2.2 DVC Environment Configuration

Next, we created a virtual namespace and configured communication for the obtained DVC. In the realized configuration, the visualization cluster and storage cluster machines at UCSD were assigned dynamic, private IP addresses for communication over the dynamically created network. As a result, the visualization cluster machines have two physical IP addresses: one for static connection to/from UvA/UIC and another for private connection to/from SOE/UCSD. To aid the management of heterogeneous resource addresses, we assigned virtual resource names to individual machines, thus allowing us to use uniform names for them. Furthermore, even though different UCSD storage clusters were selected on different runs, there was no need to reprogram the application because the resources can be assigned with consistent names.

4.2.3 Data Transfer

On the DVC, we built an interactive application that inputs the name of a dataset (a collection of scene files) from users and transfers them on-demand from remote storage clusters to the visualization

system. In our demonstration, we used 13 storage nodes at SOE, 7 nodes at UIC and 5 nodes at UvA as data sources, while using 25 visualization nodes at Calit2 as data sinks. The dataset consists of 50 files (each of size 700 MB). On each run, the application transferred these files using 25 parallel flows from data sources to data sinks; each visualization node received two files from one remote storage node. To transfer files, the application initialized the transfer on each sending and receiving node, which concurrently transferred files from remote storage disks to local disks.

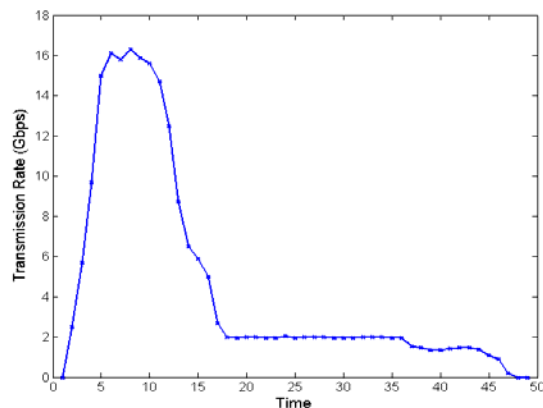


Figure 6. The trajectory of the aggregate transmission rate at the iGrid2005 visualization cluster

Figure 6 depicts the trajectory of the aggregate data transfer rate measured at data sinks at Calit2. It shows that GTP achieved a peak rate in 5 seconds of 16.3 Gbps out of the 20 Gbps link capacity, or 81.5%. There is an initialization delay representing the job synchronization and launching time. During the transfer, we observed that the 13 parallel flows between SOE and Calit2 achieved nearly the full 10 Gbps bandwidth capacity. In such case, the application is capable of sending at a rate of 13 Gbps (1Gbps on each flow), but is limited by the 10Gbps link capacity between SOE and Calit2. The aggregate rate of the 5 flows between UvA and Calit2 is approximately 4.5Gbps. These flows experienced no network limitation and achieved 90% of the application sending capability. We note that this was achieved in co-existence with background high-speed traffic (sharing the same 10 Gbps link) generated by other iGrid demonstrations. Lastly, the aggregate rate of the 7 flows between UIC and Calit2 was limited to 2 Gbps, which is shown as the “long tail” in Figure 6. Further investigation using a bandwidth measuring tool *iperf* showed that, this “long tail” behavior is due to the bandwidth limitation at the cluster at UIC. This demonstrates GTP’s capability of fully utilizing the available network capacity. For other

GTP performance measurements, the reader can refer to [10,19].

4.2.4 Data Visualization

In the final step, the transferred scene files are visualized using Fledermaus. Figure 7 illustrates the output of the tiled display during our demonstration at the iGrid2005. It shows parallel visualizations of multiple 3d theoretical models of deformation along the San Andreas Fault in California (during year 1990-2004). Each display shows a theoretical model of strain changes from large earthquakes for a given year, thus allowing researchers to see 50 years of data at the same time and interactively analyze and discuss the similarities and differences.

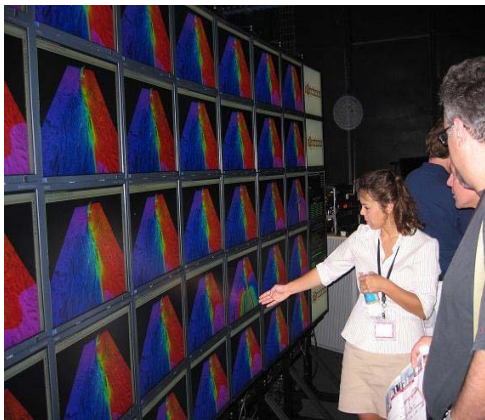


Figure 7. Parallel visualization of multiple 3D theoretical models of deformation along the San Andreas Fault in California.

4.2.5 Discussion

The visualization software Fledermaus used in the demonstration is a commercial product widely used at the Scripps Institution. Before visualizing data, it loads an entire scene file from a local disk into memory. The file loading time varies according to the file size, ranging from 2-3 seconds for a 100MB file to ~30 seconds for a 1GB file. Compared to the transfer time of the file of the same size using OptIPuter technologies (~10 seconds for a 1GB file on a 1Gbps link), the loading time presents a real bottleneck in performance of the collaborative visualization system. On the other side, if the files were transferred using a standard TCP protocol and via the shared Internet, the transfer time could have been much longer. In this case, the transfer time dominates the loading time and presents a performance bottleneck.

It should be noted that the Fledermaus software itself doesn't have built-in collaboration features such as audio and video conferencing, instant messaging, shared display or shared mouse. The key contribution

of this paper is a description and evaluation of OptIPuter middleware including a virtual computer and advanced transport protocol as enabling technologies for next-generation collaborative environments.

At the iGrid2005, we simulated a real-time working environment between scientists using high resolution tiled displays for data visualization and audio and video channels. The scientists downloaded datasets from the various storage sites during the course of their work and depended on the available videoconferencing and instant messaging tools (Polycom and iChat) to communicate with each other and to share the current state of data display. The authors concede that more sophisticated collaboration tools are required in this scenario and such tools are being developed by other members of the OptIPuter team. However, the authors also believe that an effective middleware system was put in place for this iGrid demonstration, enabling geographically distant scientists to simultaneously retrieve and display large sized datasets from remote servers while hiding the details of resource allocation and transport protocols.

5. Conclusions

In this work, we described the OptIPuter middleware which integrates end-to-end OptIPuter software technologies and presents a simple use and performance model of Lambda-Grid resources. The OptIPuter middleware enables the simple DVC abstractions, allowing an application to be conveniently constructed and effectively exploit the novel capabilities of these resources. Further, specialized transport services for group communication, such as GTP, enable applications to achieve a high aggregate throughput for parallel data aggregation from remote data sources. These capabilities are enabling technologies for emerging large-scale scientific applications, such as collaborative and remote data visualization, which involve massive collections of data objects and real-time data acquisition and visualization.

As a proof of our concept, we implemented the prototype of the OptIPuter middleware and demonstrated its benefits and capabilities to enable a collaborative visualization environment at the iGrid2005. The demonstration showed that a collaborative environment could be conveniently and effectively constructed with the OptIPuter middleware across the OptIPuter's international-scale testbed and the application was enabled to efficiently utilize the network capability over local and wide-area high-speed networks.

Acknowledgements

The authors would like to thank Cees de Laat, Paolo Grosso, Freek Dijkstra, Philip Papadopoulos, Greg Hidley, Aaron Chin, David Hutches, Praveen Kumar, Mason Katz, Sean O'Connell, Max Okumoto, Qian Lin, David Lee, Troy Chuang, Adam Burst, Tom Defanti, and Lance Long for providing end points, network, and infrastructural support at University of California, San Diego (UCSD), University of Amsterdam (UvA), and University of Illinois, Chicago (UIC). We are very grateful to Oliver Yu, Anfei Li, and Eric He for providing the PIN/PDC software. We gratefully acknowledge support from IVS for providing the licenses for Fledermaus.

The work described in this paper is supported in part by the National Science Foundation under awards NSF Cooperative Agreement ANI-0225642 (OptIPuter), NSF CCR-0331645 (VGrADS), NSF ACI-0305390, and NSF Research Infrastructure Grant EIA-0303622. Support from the UCSD Center for Networked Systems, Nortel, Force10, Cisco, BigBangwidth, and Fujitsu is also gratefully acknowledged.

References

- [1] T. DeFanti, et al., "Teleimmersion and Visualization with the OptIPuter," in *Proceedings of the 12th International Conference on Artificial Reality and Telexistence (ICAT2002)*, December, 2002.
- [2] J. Laude, "DWDM Fundamentals, Components, and Applications," Artech House, January, 2002.
- [3] I. Foster and C. Kesselman, editor, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, 1999.
- [4] L. Smarr, A. A. Chien, T. DeFanti, J. Leigh, and P. Papadopoulos, "The OptIPuter," in *Communication of the ACM*, 46(11), November, 2003. www.optiputer.net.
- [5] LambdaVision.
<http://www.evl.uic.edu/cavern/lambdavisio/>
- [6] A. A. Chien, et al., "OptIPuter System Software Framework," UCSD Technical Report CS2004-0786, 2004.
- [7] K. H. Kim, "Wide-area Real-time Computing in a Tightly Managed Optical Grid – An OptIPuter Vision," in *Proceedings of the 18th IEEE International Conference on Advanced Information Networking and Applications*, March, 2004.
- [8] N. Taesombut and A. A. Chien, "Distributed Virtual Computer: Simplifying the Development of High-Performance Grid Applications," in *Proceedings of the Grids and Advanced Networks Workshop (GAN04)*, April, 2004.
- [9] C. Liu and I. Foster, "A Constraint Language Approach to Matchmaking," in *Proceedings of the 14th International Workshop on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE'04)*, March, 2004.
- [10] J. Leigh, et al, "An Experimental OptIPuter Architecture for Data-Intensive Collaborative Visualization," in *Proceedings of the 3rd Workshop on Advanced Collaborative Environments*, 2003.
- [11] X. Wu and A. A. Chien, "GTP: Group Transport Protocol for Lambda-Grids," in *Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'04)*, April, 2004.
- [12] C. Xiong, et al, "LambdaStream – A Data Transport Protocol for Network-Intensive Streaming Applications over Photonic Networks," Extended Abstract for the 3rd International Workshop for Protocols for Fast Long Distance, February, 2005.
- [13] D. Katabi, M. Handley, and Charles Rohrs, "Internet Congestion Control for Future High Bandwidth-Delay Product Environments," in *Proceedings of ACM Sigcomm 2002*, August, 2002.
- [14] Y. Gu and R. Grossman, "SABUL: A Transport Protocol for Grid Computing," in *Journal of Grid Computing*, vol 1, 2004.
- [15] R. Grossman, et al, "Teraflows over Gigabit WANs with UDT," in *Journal of Future Generation Computer Systems*, 21(4), April, 2005.
- [16] E. He, J. Leigh, O. Yu and T. DeFanti, "Reliable Blast UDP: Predictable High Performance Bulk Data Transfer," in *IEEE Cluster Computing*, 2002.
- [17] O. Yu, "Intercarrier Interdomain Control Plane for Global Optical Networks," in *Proceedings of IEEE ICC*, June, 2004.
- [18] E. He, J. Alimohideen, J. Eliason, N. K. Krishnaprasad, J. Leigh, O. Yu, and T. A. DeFanti, "Quanta: A Toolkit for High Performance Data Delivery over Photonic Networks," in *Journal of Future Generation Computer Systems*, 19(6), August, 2003.
- [19] L. Gommans, C. D. Laat, B. V. Oudenaarde and A. Tall, "Authorization of a QoS Path Based on Generic AAA," in *Journal of Future Generation Computer Systems*, 19(6), August, 2003.
- [20] E. He, Photonic Domain Controller. www.evl.uic.edu/cavern/rg/20031003_he.
- [21] X. Wu and A. A. Chien, "A Distributed Algorithm for Max-min Fair Bandwidth Sharing," UCSD Technical Report, 2006. www-csag.ucsd.edu/papers/ryan_dist.pdf
- [22] X. Wu and A. A. Chien, "Evaluation of Rate-based Transfer protocols for Lambda-Grids," in *Proceedings of the 12th IEEE International Symposium on High-Performance Distributed Computer (HPDC)*, June, 2004.
- [23] L. Smarr et al, "The OptIPuter, Quartzite, and Starlight Projects: A Campus to Global-Scale Testbed for Optical Technologies Enabling LambdaGrid Computing," in *Proceedings of OFC/NFOEC*, 2005.
- [24] D. Oppenheimer, J. Albrecht, D. Patterson, and Amin Vahdat, "Scalable Wide-Area Resource Discovery," UC Berkeley Technical Report UCB//CSD-04-1334, July 2004.
- [25] T. Erlebach and K. Jansen, "Scheduling of Virtual Connections in Fast Networks," in *Proceedings of the 4th Parallel System Algorithms Workshop*, 1996.
- [26] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, "Optimization by Simulated Annealing," Science No. 220, pp671-680, 1983.

