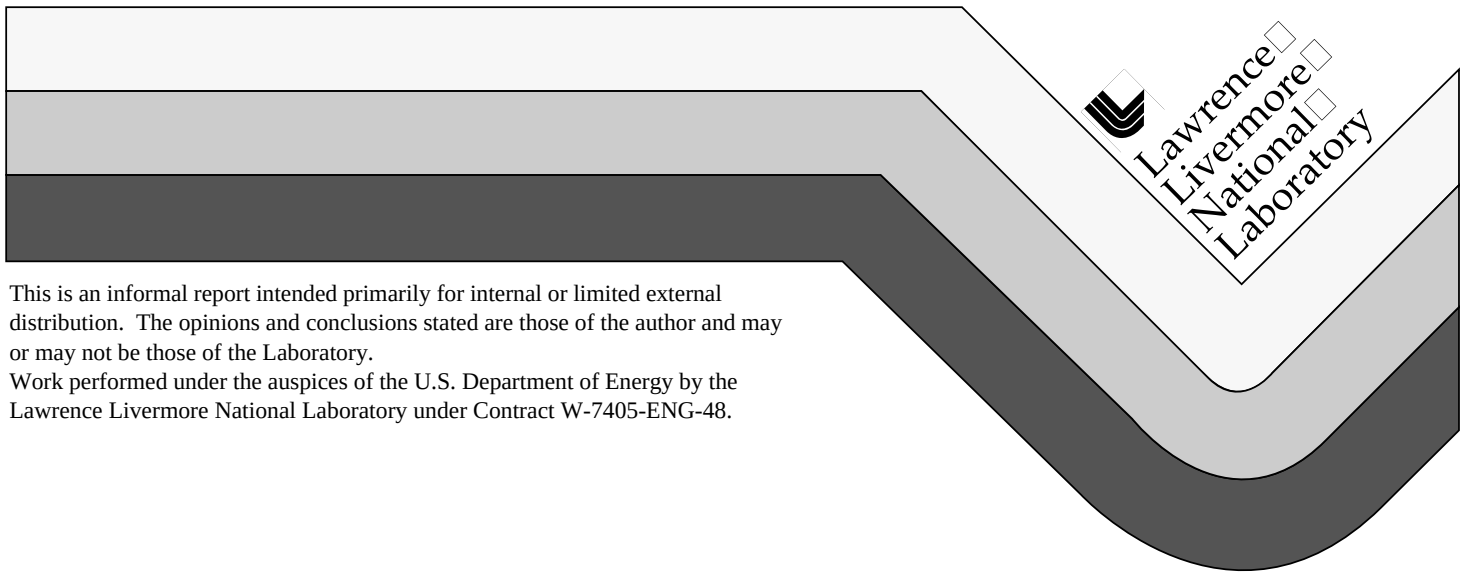
UCRL-ID-130896

Modeling and Analyzing Visualization Post-Processing Over Distance

Dave P. Wiltzius

5/21/97



Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

Modeling and Analyzing Visualization Post-processing over Distance

Dave Wiltzius
Lawrence Livermore National Lab
wiltzius@llnl.gov

May 21, 1997

1. Functional Decomposition with Data Flows

Stockpile stewardship requires a high-end computing capacity complemented with a balance of memory capacity and bandwidth, interconnect bandwidth, local and global disk capacity and bandwidth, network bandwidth, and archival capacity and bandwidth. This appendix will provide a detailed analysis that will identify technical issues arising from various user interactions with a computer with a peak capacity of 10 TFLOPs and with 5TB of memory.

1.1 Identifying a User Interaction Stressful to Computing Resources

The focus of this technical analysis will be on user interactions that are most stressful to such a computer facility (i.e., the computer, interconnect, I/O and archival system, and networks). The user interaction selected for analysis is the visualization post-processing of large problem results. However, execution of the application can include some steps in the visualization post-processing, and hence is included in the analysis in this study.

1.2 Impact of Distance on High-End Computing

Of specific interest are technical issues that arise when the high-end computing resources are located at a considerable distance (not accessible via a traditional local-area network (LAN)) from the users and hence must be accessed over a wide-area network (WAN). This distance computing configuration will consist of two types of facilities:

1) The single high-end computing facility which includes the 10TFLOPs machine with 5TB of memory and associated computational resources (global disk, archival storage, etc.).
2) The remote sites with users and some set of computational resources (presumably at least a desktop workstation for each user).

The analysis will create a baseline configuration where users are local to the high-end computing facility. Next a distance will be introduced between the high-end computing facility and the users at remote sites. Distance between computing components introduces two primary technical factors: latency (at minimum, that of speed of light) and bandwidth. The latter cannot be ignored since it is severely constrained by technical feasibility (in 1997 a long distance fiber is limited to about 5GB/s) and cost (in 1997 OC-12c WAN bandwidth or 622Mb/s (~67MB/s payload) is about $3M per year per individual connection).

Several remote site configurations will be defined providing additional computing resources, local disk, and archival storage to mitigate latency and bandwidth issues. The technical analysis will assess and compare the remote site configurations using the following as metrics:

1) The local-area network, wide-area network and I/O bandwidths required to support the remote site configuration (i.e., technical feasibility),
2) The latency experienced by the remote user, as compared to the baseline system (i.e., anticipated impact on user productivity),
3) The cost of the remote site configuration.

The first step is to select a methodology appropriate for the analysis of user interactions with a high-end computing system and associated computational resources. This methodology will be used to identify issues, and help develop, assess and compare distance computing configurations.

1.3 Methodology: Functional Decomposition of Data Flow

The methodology chosen borrows from modern system engineering practices dealing with complex hardware and software systems [1], and proceeds as follows:

1) The computing system is decomposed into functional elements for each type of user interaction (e.g., post-processing visualization). This results in a model for each type of user interaction and will be the framework for enumerating the technical issues.
2) Significant data flows between functional elements are identified in this model (figure 1). Latency and bandwidth numbers will ultimately be derived from the data flows.
3) The model's functional elements are mapped onto hardware components, initially resulting in a baseline configuration (figure 2) with all components and users being local. This mapping is necessary to make the numbers derived from the data flows meaningful (for example, writing to memory is a functional data flow, but the latency and bandwidth depends upon whether or not the memory location is in cache).
4) User and simulation requirements, and code characteristics are inserted into this model by characterizing the workload on the system with different time and event driven data flow requirements. This will provide, for example, data set sizes and the frequency that data sets are transferred.
5) Latencies and bandwidths between functional, and hence hardware elements are determined for single-user and multi-user workloads.
6) With an understanding of the simulation requirements, the various functional elements, and hardware components, several remote site configurations are developed for the distance computing situation.
7) Each configuration for a remote site is evaluated, in a technical context, considering: technical feasibility (bandwidth), anticipated impact on user productivity (latency), and cost.

2. Visualization Post-processing Model

The visualization post-processing model (figure 1) characterizes the user interaction with the computing facility when visualizing timestep data sets as they are generated. This

same model will be used to analyze the impact on the user when a WAN is introduced with several configurations of remote sites. These configurations will consist of various computing resources (SMPs, archival storage, local disk, rendering engine in addition to the desktop) local to the user for the purpose of hiding latency, reducing the required bandwidth of the WAN, and otherwise optimizing the productivity of the user.
The analysis will proceed by applying the methodology described above to the visualization post-processing model.

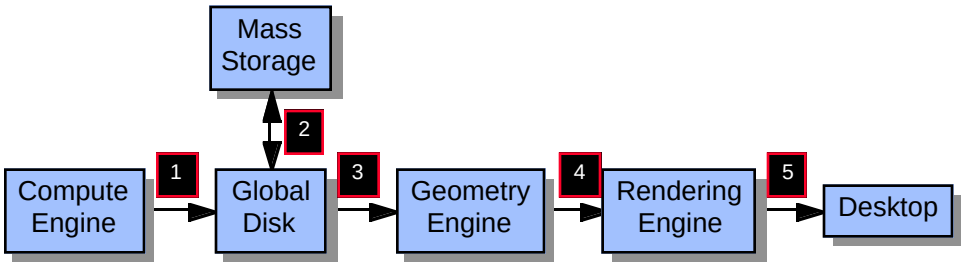


Figure 1: Visualization Post-processing Functional Model with Data Flows

2.1.1 Functional Decomposition and Data Flow Identification

This model requires six functional elements: Compute engine, global disk, archival storage, geometry engine, rendering engine, and user desktop. The visualization post-processing model is represented with five data flows, enumerated below and identified on the figures in this section as 1-5:

1) The application is running on the *compute engine* (clustered SMP) and generating timestep datasets which are written to the machine's *global disk*. The timestep datasets consist of restart datasets and visualization (movie) datasets. The restart datasets are 0.1 to 1.0 of the memory used by the application, and are infrequently written. The visualization datasets are about 0.1 to 0.01 the size of the memory used by the application and are written more frequently than the restart datasets.

   The application has two options when writing these datasets. The first option is to use double buffering so that the dataset can be written (via a separate thread) while the calculations proceed (in another thread). For this option it is desirable that the datasets be written before the next timestep is calculated.

   Some applications require most of the memory so that double buffering is not possible. This is more likely for the restart datasets since they are capturing the state of the application and hence are much larger than the visualization dataset. So the application exercises the second option, which is to stall further calculations and wait for the datasets to be written. In this it is most desirable for the datasets to be written as quickly as possible since the high-end computing resources are effectively idle.

2) Since space on the *global disk* must be available for other users, the timestep datasets are migrated from global disk to the archive system (e.g., HPSS) while the application is running.

3) For post-processing visualization, the user will view a recent timestep visualization data set. The *geometry engine* function is done by software that randomly accesses the visualization data set to extract the sub-domain data set (e.g., iso-surface), and then reduces the sub-domain data set to a three-dimensional geometry data set for the surface.

4) The geometry dataset is written to the *rendering engine*, which then renders the image as a 3MB frame dataset for the 1Kx1Kx24 bit raster image (i.e., a square frame 1024 pixels on a side, and with 24 bits of color information for each pixel).

5) The 3MB frame dataset is sent by the rendering engine over the network to the *user's desktop*. At minimum, the user would like 3 frames per second (fps). Practically, the user rarely needs more than 10fps.

   Additionally, the user may which to view the surface from different perspectives ("fly-around"). Since the rendering engine has the three dimensional geometry for the surface, the user interacts just with the rendering engine to perform the fly-around.

2.1.2 Creating the Baseline Configuration

The visualization post-processing model has now been functionally decomposed, and the data flows identified and enumerated. Next the functional elements will be mapped onto hardware to generate the baseline configuration (figure 2). The compute engine functional element and global disk functional element are naturally mapped to the clustered SMP and its global disk, respectively.

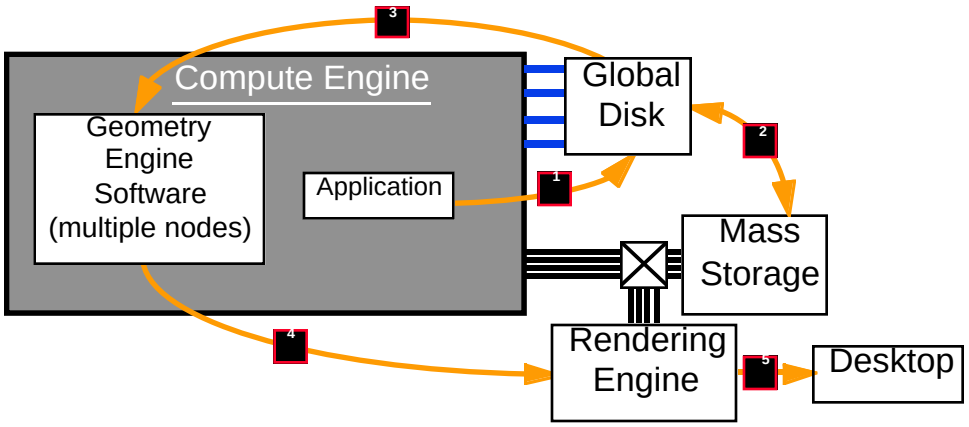


Figure 2: Baseline Configuration - Visualization Post-processing Model w/Data Flows

As noted above, a recent timestep dataset is accessed from the global disk by the software geometry engine. In general, the global disk I/O bandwidth on the clustered SMP is significantly greater than the network bandwidth off the clustered SMP, or the I/O bandwidth from the archive storage. Hence the geometry engine application is elected to run on a few nodes of the clustered SMP.

Finally, the baseline configuration is completed by mapping the rendering engine functional element on a rendering engine (e.g., an SGI Infinite Reality system), and the desktop functional element to a desktop workstation.

2.1.3 Workload Data Sets

To continue with the analysis of the visualization post-processing model, it is required to characterize the problem sets running on the clustered SMP. This is done by extrapolating workload characteristics from current ASCI applications and their problem sets.

The discussion that follows will refer to 5 data sets for the visualization post-processing model:

1) The restart data set (part of data flows 1 and 2), which generally does not contribute to this analysis.
2) The visualization data set (the remainder of data flows 1 and 2), which is the data set most frequently generated by the application, and is of interest to this model.
3) The sub-domain data set (data flow 3), which is extracted from the visualization data set by the geometry engine application to gather (for example) the three dimensional physical data for an iso-surface. The geometry engine is the first application in the visualization post-processing pipeline (figure 3).
4) The geometry data set (data flow 4), which is the three dimension graphical representation of the sub-domain data set. For example, the geometry data set could be the coordinates and graphical attributes for the vertices of triangles.
5) The frame data set (data flow 5) is a raster image whose size is determined by the number of pixels to be displayed, and the number of colors per pixel. A typical frame data set size is 3MB for a 1Kx1K frame with 24 bits of color (8 bits each for red, green, and blue).

2.1.4 Visualization Pipeline

The visualization post-processing interaction establishes a visualization pipeline that starts with a visualization data set and ends with the user interacting with the rendering engine to perform a fly-around of the surface data. To start a fly-around, the timestep sub-domain data set is extracted from the visualization data set by the geometry engine. The geometry engine creates the geometry data set and sends it to the rendering engine. The geometry data set is received by the rendering engine, and then the user interacts just with the rendering engine to perform the fly-around, updating the screen at 10-30fps. Hence traversing the visualization pipeline creates data flows 3 through 5, but the user interaction is focused on data flow 5 (figure 3).

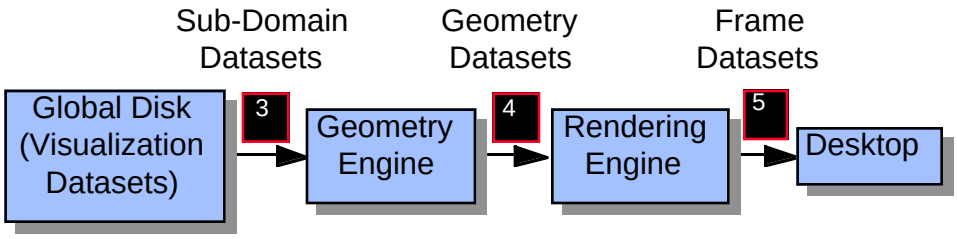


Figure 3: Visualization Pipeline w/Data Flows

When the user reaches a boundary of the surface, a timestep visualization data set must again be accessed to retrieve the next sub-domain data set. The visualization pipeline is again traversed until the screen is refreshed from the new geometry data set. The user is willing to wait up to 30 seconds for this screen refresh before continuing with the fly-around.

2.1.5 Analysis Strategy and Corresponding Assumptions

The goal of this analysis is to determine a distant (i.e., over a WAN) high-end computing configuration that is reasonable: minimally impacts the users' productivity, and at a reasonable cost.

Presently the user runs the application, which will generate and archive the restart and visualization data sets (or just the former, which will then also be used for visualization post-processing). At some later time, the user retrieves the visualization data sets and analyzes the results using visualization.

Some considerations and scenarios are identified below which will assist in achieving the goal of this analysis.

2.1.5.1 Bandwidths Motivated by Shared Resources (Global Disk)

The application running on the distant high-end computer writes the restart and visualization data sets to the global disk. The global disk is a shared resource. Specifically, when an application begins executing it will likely assume there is a considerable portion of the global disk available. Hence as an application writes to the global disk, its strategy should be to migrate the data sets to archive storage at about the same rate as they are being written to global disk.

Failure to do this could result in filling the global disk, forcing the application to idle while global disk resources are migrated to archive storage. This idles the high-end computer.

Another possibility is that the application completes before the global disk gets filled, but there is not enough global disk available for the next application. Again, an application must idle until the migration of data sets to archival storage frees sufficient global disk space.

Assumption

To ensure that the global disk has enough available space when an application completes, the visualization and restart data sets will be migrated to archival storage at about the same rate that they are generated. The rate of generating the visualization data sets corresponds closely to the timestep calculation time. Hence one set of bandwidths for each of the data flows can be determined from the timestep calculation time.

2.1.5.2 Data Set Migration over the WAN

For distant high-end computing, one of the five data flows identified above (figure 1) would have to occur over the WAN. So this analysis will consider doing some of the

visualization post-processing as part of the application run. For example, the first scenario will migrate the visualization data set over the WAN to the remote user's site. The next scenario will extract the sub-domain data set from the visualization data set as part of the application run, and migrate the sub-domain data set over the WAN. Ultimately, the application run could a priori perform the entire visualization pipeline, and just migrate the raster images over the WAN to the user's remote site. These scenarios have been developed below, and result in remote site configurations A through D which consider the migration of the visualization data set (configuration A), sub-domain data set (configuration B), geometry data set (configuration C), and raster data set (configuration D).

For this aspect of the analysis, the user may be concerned about the latency incurred from the time that the data set is generated, and the time that it is available at the user's remote site. Since the migration of a data set over the WAN can occur while the next time step is being calculated, the latency is calculated only for one data set (rather than waiting for all data sets to be generated, then waiting for all data sets to be migrated to the remote site).

Assumption
The migration of data sets will occur while the application is executing, and will begin as soon as the data set is written to global disk. For the sub-domain, geometry, and raster data sets this also assumes that the resources will be available to perform the appropriate parts of the visualization post-processing to generate the data set.

Assumption
For the sub-domain, geometry, and raster data sets, the user knows a priori which data is of interest. Hence the application can be scripted to generate the appropriate data sets (i.e., which physics to visualize, which perspective, etc).

Assumption
An a priori knowledge of what is of interest will allow scripting of the application run so that some of the post-processing can be done at the distant high-end computing site.

2.1.5.3 User Interacting with Resultant Data Set

Visualization post-processing will begin with the user viewing a data set. The user interacts with the local visualization hardware to do the fly-around. When the end of the surface is reached, the user must depend upon the distant high-end computer to retrieve the appropriate visualization data set to allow the post-processing to resume.

The distant high-end computing site, the visualization data set must be readily available so that the sub-domain data set can be extracted to proceed with the visualization pipeline (figure 3). The appropriate data set is then transferred to the remote site over the WAN.

This scenario does not apply if the visualization data sets are migrated to the user's site since then all post-processing information is local to the user.

If the raster data set is migrated to the user's site, then the entire post-processing session must be fully scripted a priori by the user (i.e., the result is a movie).

Assumption
The user will tolerate a 30 second delay in this scenario, waiting for the fly-around to resume. This time interval allows another set of data flow bandwidths to be determined. Note that the bandwidths can arbitrarily large by decreasing the 30 second maximum waiting period of the user.

2.1.5.4 Bandwidth and Latency for Interactive User (Debug and Steering)

The application running on the distant high-end computer writes the restart and visualization data sets to the global disk. In this scenario, the user is interacting with the running application. At some point, the user wishes to view the latest result. Instead of a user working with the data sets that have been migrated to the local site, the user is actually waiting for the post-processing to occur as the timestep completes.

The could be done in conjunction with the migration of data sets. In that case, the data sets would arrive at the user's site, and the user would being analyzing the results. The latency from the time that the timestep completes and the data set is available at the user's site would be an issue. It is assumed that the user would tolerate a delay less than the timestep calculation time.

The other issue would be identical to the scenario in the previous section (User Interacting with Resultant Data Set) when the fly-around goes beyond the boundary of the data set being viewed.

Assumption
The user will expect the most recent timestep data set before the next timestep has been calculated. For long timestep calculation times, is very likely to be limited much further (say, to a few minutes after the visualization data set is written).

Assumption
While doing a fly-around and the boundary of the data set has been reached, the user will wait up to 30 seconds for the next data set to appear at the desktop.

Alternatively, the entire visualization pipeline could be performed at the distant high-end computing site and only the raster images sent to the user's site (this may be reasonable for a debug session, for example).

2.1.5.5 Network Bandwidths

Network bandwidth is inconsistently quoted by industry. What is of real interest is the actual payload bandwidth; that is, the bandwidth available to the application (and hence the user).

The actual network payload bandwidth varies with the platform and its version of the operating system, network hardware (e.g., routers, switches), and many other factors.

This makes it impossible to make any general statements about the actual network payload bandwidth. All else being equal, the actual payload bandwidth may be quite different between two versions of an operating system for example.

Statements can be made about the maximum theoretical network payload bandwidth. Still, care must be taken. The maximum theoretical network payload bandwidth varies greatly with the packet size, for example. Smaller packets have more overhead, and larger packets have smaller overhead which results in a larger maximum theoretical network payload bandwidth. Usually the latter is quoted, of course.

Next, the media bandwidth may be quoted. This bandwidth may include other overhead which drastically inflates the bandwidth. With ATM, for example, the bandwidths are often quoted as a multiple of OC-1 (51.84Mb/s). This bandwidth includes the overhead from SONET (~3%) and ATM headers (~9%). Hence for OC-3 ATM (155Mbs/), the maximum theoretical network payload is about 136Mb/s. For Fibre Channel, the bandwidth includes the data encoding so a 1Gb/s Fibre Channel link has a maximum theoretical network payload bandwidth of less than 800Mb/s.

Assumption
The network payload bandwidth specified in this report is the maximum theoretical network payload bandwidth. There are clearly identified occasions where a specific instance of the actual network payload bandwidth is used.

2.1.5.6 Annual Cost Adjustments

Moore's Law suggests that the performance of processors will double every 18 months. This has been used to conclude that the purchase power for computational capacity will double every 18 months. Similar trends (but not identical, nor related to Moore's Law) exist for the costs of memory, disk, tape, and WANs. These are the primary components of all computing resources considered in this report.

For example, a visualization server consists of a large amount of memory, processing elements, and disk. One unique component is the rendering engine (e.g., Infinite Reality engine on an SGI).

Assumption
Costs of memory, disk, tape, and processing capacity doubles about every 18 months. Many of these are recognized trends. The cost of WAN capacity appears to double about every 4 years. All of these trends are subject to technical and consumer market influences.

2.1.5.7 Workload 1: Single "Hero" User (Capability), from LLNL Data

The model will first be analyzed using the workload characteristics of the capability user: A single "hero" user utilizing the entire machine. Next the model will be analyzed with a multi-user workload.

The first single user workload characterization for a 10TFLOP machine with 5TB of memory will be a 1 billion zone hydrodynamics problem with 500 bytes of physics data

per zone (the mean of several physics codes used today). The restart datasets will be 500GB and the visualization datasets will be about 100GB. The restart datasets will be generated twice per hour while the visualization datasets will be generated once every 150 seconds. The problem will run for almost 21 hours, resulting in 500 visualization datasets and 40 restart datasets. A run for this workload results in 20TB from the restart datasets and 50TB from the visualization datasets for a total of 70TB of datasets stored on global disk and migrated to archival storage.

Bandwidths that are realized today will be used in the baseline configuration (figure 2) to calculate the latencies experienced by the customer.

The bandwidth for data flow 1 (between the nodes of the clustered SMP and the global disk) should be as great as possible since it is assumed that the application must write the visualization dataset before it can proceed with the calculations for the next timestep. A reasonable guideline established by the customers is that the I/O time will not exceed 10% of the time to calculate the timestep. If we assume an I/O bandwidth over the cluster interconnect of 6GB/s then the 100GB visualization dataset will be written in about 17 seconds, which is slightly greater than 10% of the 150 second timestep calculation time but appropriate for the customers guidelines.

Data flow 2, from global disk to the archive storage via the SMP I/O nodes is concurrent with next timestep calculation and hence no latency is incurred. However, to ensure that the application will never stall due to the global disk being filled to capacity, the migration of datasets from global disk to the archive should occur at the same rate that the datasets are being created. That is, the datasets created for each timestep should be migrated to archive within the timestep calculation time. For a 100GB dataset created during a 150 second timestep calculation, the migration to the archive should occur at 667MB/s (or 8 HIPPI-800 stripes at 83MB/s each - considerably faster than what can be obtained today on many machines).

Now the latency for the visualization pipeline (data flows 3-5) will be determined. Again, this is the time from the user request for a new image (in this discussion, as soon as the latest timestep completes), and when the image appears on the user's desktop.

The sub-domain data read by the geometry engine (data flow 3) is again assumed to be the clustered SMP interconnect bandwidth, so the global disk reads occur at 6GB/s. The geometry engine application only reads the desired sub-domain dataset, for example the iso-surface of the density variables. For a 100GB visualization dataset, it is expected that this sub-domain dataset would be 3GB. Hence it would require 0.5 seconds to read in the sub-domain data plus a few milliseconds to calculate the 8 million triangle 400MB three dimensional surface geometry dataset.

In the baseline configuration for this model, the geometry dataset is the first data that leaves the clustered SMP (data flow 4). The 400MB geometry dataset is sent to the rendering engine. With a 4 way stripe of HIPPI-800 at 40% efficiency (=160MB/s) this would take 2.5 seconds.

The rendering engine would, with the appropriate software developed, produce a 3MB uncompressed (compression is considered in Appendix A) raster frame in a few milliseconds. This 3MB frame dataset would then be delivered to the user's desktop (data flow 5). With HIPPI-800 at 40% efficiency (=40MB/s) this would take 75 milliseconds.

Therefore, in the baseline configuration with the user being local to the capability computing resources, the user would be able to view the timestep surface image within 3.1 seconds after the timestep data was written to global disk by the SMP cluster.

These numbers appear in the tables below as row "Wrkld 1."

2.1.5.8 Workload 2: Single "Hero" User (Capability), from LANL Data

The LANL single user data describes a problem that takes 100 hours to run, and will generate 5 restart datasets and 100 visualization datasets. Each restart dataset is equal to 1.0 the size of the memory or 5TB. Each visualization dataset is 0.01 of the memory size or 50GB. The datasets are produced at a uniform and the restart datasets can be ignored in this analysis.

The bandwidth for data flow 1 (between the nodes of the clustered SMP and the global disk) should be as great as possible since it is assumed that the application must write the visualization dataset before it can proceed with the calculations for the next timestep. If we assume an I/O bandwidth over the cluster interconnect of 6GB/s then the 50GB visualization dataset will be written in about 8.3 seconds or about 0.2% of the 1 hour timestep calculation time. However, a 5TB restart dataset will take over 833 seconds (13.9 minutes) to write at 6GB/s, but this is done only once every 20 hours.

Data flow from global disk to the archive storage via the SMP I/O nodes is concurrent with next timestep and hence no latency is incurred. However, to ensure that the application will never stall due to the global disk being filled to capacity, the migration of datasets from global disk to the archive should occur at the same rate that the datasets are being created. That is, the datasets created for each timestep should be migrated to archive within the timestep calculation time. For a 50GB dataset created during a 1 hour timestep calculation, the migration to the archive should occur at 14MB/s (a bandwidth achievable on many machines).

The latency for the visualization pipeline (data flows 3-5) will be determined.

The sub-domain data read by the geometry engine (data flow 3) is again assumed to be the clustered SMP interconnect bandwidth, so the global disk reads occur at 6GB/s. The geometry engine application only reads the desired sub-domain dataset, for example the iso-surface of the density variables. For a 50GB visualization dataset, it is expected that this sub-domain dataset would be 1.5GB. It would require 0.25 seconds to read in the sub-domain data plus a several milliseconds to calculate the 8 million triangle 400MB three dimensional surface geometry dataset.

In the baseline configuration for this model, the geometry dataset is the first data that leaves the clustered SMP (data flow 4). The 400MB geometry dataset is sent to the rendering engine. With a 4 way stripe of HIPPI-800 at 40% efficiency (=160MB/s) this would take 2.5 seconds.

The rendering engine would, with the appropriate software developed, produce a 3MB uncompressed (compression is considered in Appendix A) raster frame in a few milliseconds. This 3MB frame dataset would then be delivered to the user's desktop (data flow 5). With HIPPI-800 at 40% efficiency (=40MB/s) this would take 75 milliseconds.

Therefore, in the baseline configuration with the user being local to the capability computing resources, the user would be able to view the timestep surface image within 2.83 seconds after the timestep data was written to global disk by the SMP cluster.

The numbers appear in tables below as row "Wrkld 2."

2.1.5.9 Workloads 3-6: Multi-User (Capacity), from LANL Data

Workload 3 is very similar to workload 2. However, it is assumed that the problem is about half the size since only half the machine is used. Hence the sub-domain dataset (data flow 3) and the geometry dataset (data flow 4) are half the size.

For workloads 4-6, similar scale factors are applied so that the datasets are proportional to the problem size, which is assumed to be proportional to the fraction of the machine that is used by the problem.

The visualization pipeline latencies for the baseline configuration (all computational components local to the user) appear in table 1 below as rows "Wrkld 1" through "Wrkld 6." The visualization pipeline latency of 3.1 seconds for workload 1 is largest.

Type of Run	Latency (sec) for Data Flows 3 (6GB/s) + 4 (160MB/s) + 5 (40MB/s)	Max Num of Users
Wrkld 1, LLNL: High End 3D	0.5 + 2.5 + 0.075 = 3.08s	1
Wrkld 2, LANL: High End 3D	0.25 + 2.5 + 0.075 = 2.83s	1
Wrkld 3, LANL: Full 3D	0.125 + 1.25 + 0.075 = 1.45s	2
Wrkld 4, LANL: Scoping 3D, High Res 2D	0.025 + 0.125 + 0.075 = 0.23s	10
Wrkld 5, LANL: Full 2D	0.005 + 0.025 + 0.075 = 0.11s	50
Wrkld 6, LANL: Scoping 2D	0.00 + 0.001 + 0.075 = 0.08s	100

Table 1: Summary of Single User Baseline Latencies

2.2 Developing a Remote Site Configurations

To identify and assess distance computing issues it will be necessary to split the baseline configuration into two sites separated by a WAN. The local site hosts the high-end computer with local and global disk storage, archival storage, and a rendering engine. The

remote site will consist of, at minimum, users and a desktop workstation. To hide latencies and manage the bandwidth requirements of the WAN, remote site configurations will be considered that additionally include disk storage, archival storage, some computational resources (e.g., an SMP), and a rendering engine.

Extending the visualization post-processing model to include a remote site will begin by reverting back to the functional elements (figure 1). This is necessary since the mapping to hardware will be performed again to define the remote site configuration.

There are four interesting remote site configurations identified during this study:

1) Configuration A (figure 4): The restart and visualization datasets are written to the global disk (data flow 1) at regular intervals, with the restart dataset being written much less frequently than the visualization dataset. While the calculations proceed with timestep datasets written to the global disk, another application will be migrating the timestep datasets from the global disk to the mass storage system (data flow 2) for archival.

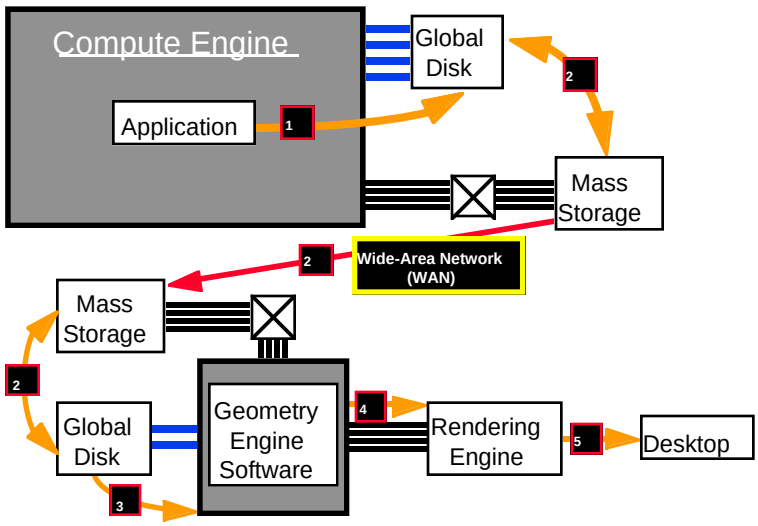


Figure 4: Configuration A - Remote Site Configured for Migrating Visualization Dataset over WAN

In configuration A it is possible for an application (or script) to archive the visualization datasets (data flow 2) at the remote site over the WAN. The time constraints imposed by the user will determine the bandwidth required of the WAN. For example, with all workloads the last visualization dataset will arrive within 25 minutes (Table 2) after being written to global disk with an OC-12c WAN.

This will give the remote users maximum flexibility and freedom when performing visualization post-processing. The remote site would consist of archival storage to cache the visualization datasets, an SMP with local disk (for the geometry engine functional element), a rendering platform, and a desktop (see figure 4 - the numbers identify the data flows).

Type of Run	Max Num Users	Timestep Calculation Time (sec)	Latency for one vis dataset	Latency for one sub-domain dataset	Latency for one geometry dataset	Latency for one frame
Wrkld 1, LLNL: High End 3D	1	150	24.9m	44.8s	6.0s	0.04s
Wrkld 2, LANL: High End 3D	1	3600	12.4m	22.4s	6.0s	0.04s
Wrkld 3, LANL: Full 3D	2	1800	6.2m	11.2	6.0s	0.04s
Wrkld 4, LANL: Scoping 3D, High Res 2D	10	360	1.2m	2.2s	6.0s	0.04s
Wrkld 5, LANL: Full 2D	50	36	15.0s	0.4s	6.0s	0.04s
Wrkld 6, LANL: Scoping 2D	100	18	0.7s	0.02s	6.0s	0.04s

Table 2: Latency for one dataset at maximum OC-12c rate (67MB/s)

2) Configuration B (figure 5): A priori, extract the sub-domain dataset from the visualization dataset and migrate the sub-domain datasets (data flow 3) over the WAN to the remote site. Now the user has less flexibility, but is still able to locally do a fly-around for the entire sub-domain surface. The remote site configuration would consist of an SMP with local disk (for the geometry engine), a rendering engine, and a desktop.

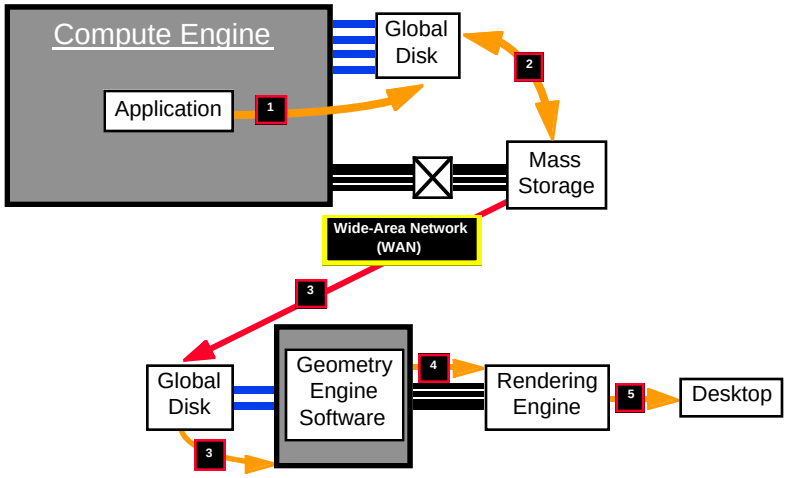


Figure 5: Configuration B- Remote Site Configured for Migrating Sub-Domain Dataset over WAN

When migrating the sub-domain datasets to the remote site over an OC-12c WAN, all workloads the last visualization dataset will arrive within 45 seconds (Table 2) after being written to global disk.

3) Configuration C (figure 6): Use the geometry engine functional element to reduce the sub-domain dataset to the geometry dataset (data flow 4), which is a portion of the three dimensional surface. The geometry dataset traverses the WAN to the remote

site. The user can locally do a fly-around, but when the boundary of the surface is encountered the user must utilize the high-end computing facility to retrieve the sub-domain dataset from the visualization dataset. The remote site configuration would consist of a rendering engine, and a desktop.

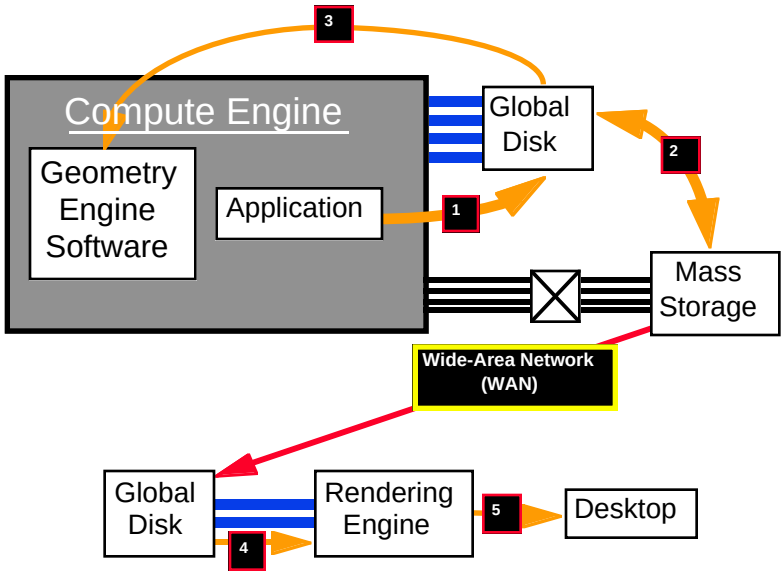


Figure 6: Configuration C - Remote Site Configured for Migrating Geometry Dataset over WAN

When migrating the 400MB geometry datasets to the remote site over an OC-12c WAN, all workloads the last visualization dataset will arrive within 6 seconds (Table 2) after being written to global disk.

4) Configuration D (figure 7): Use the high-end computing facility to perform the entire visualization pipeline, which would result in a 3MB (uncompressed) frame dataset. The remote site configuration would consist of a desktop.

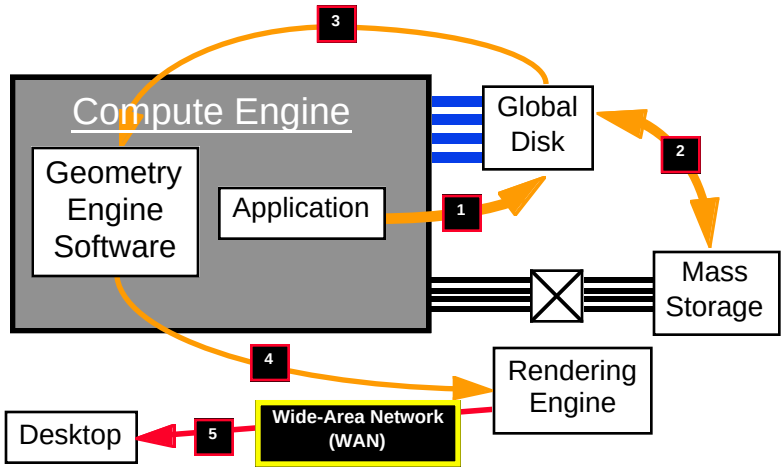


Figure 7: Configuration D - Remote Site Configured for Viewing Frame Dataset over WAN

When migrating the 3MB frame datasets to the remote site over an OC-12c WAN, all workloads the last visualization dataset will arrive within 40 milliseconds (Table 2) after being written to global disk. For comparison, the time-of-flight latency over ESnet between LLNL and LANL is presently about 33 milliseconds.

In Table 3 below, the timestep dataset size and maximum multi-user bandwidths are calculated for each workload. The visualization dataset size is determined by the workload type. Several of the numbers that immediately follow, and used in Table 3, are approximate and derived from the LLNL single user workload. The reduction factor from the visualization dataset to the sub-domain dataset (e.g., iso-surface) is assumed to be 33.3. The geometry dataset for this size problem is about 400MB, and the frame dataset is fixed at 3MB and is delivered at 10fps (=30MB/s) for each user.

The first column in Table 3 is the number of users running this workload that would fully utilize the machine.

The next column is the wall-clock calculation time for each timestep.

The last 4 columns (labeled Config A-D) have 4 numbers. Recall that the user flexibility in visualization post-processing is greatest for configuration A (figure 4, entire visualization dataset cached at remote site) and gradually decreases with the least flexibility in configuration D (figure 7, raster frames sent to remote site).

The first number is the size of the data set. The next two numbers are based on the maximum amount of data generated per timestep, which is the maximum number of users (second column) times the size of the data set. These numbers for each workload (i.e., row) is assuming that the machine is consumed by homogenous workload.

The second number, the timestep driven bandwidth, is determined by the rate at which the data sets from all users are created, which is the timestep calculation time. (column 3). It is desired that the data sets are received at the remote site at least as fast as they are created. If this were not the case, then access to the latest timestep data set from the remote site would lag behind more as the run progresses.

The third number is the bandwidth required to transfer the data set within 30 seconds. This time interval is the maximum length of time that the user would be willing to wait for the next data set when reaching the boundary of the current view during a fly-around.

The fourth number is the latency in seconds (“s”) or minutes (“m”) to migrate the dataset for all users on the system. For example, in row 3 for workload 3, the machine can accommodate at maximum 2 users. Each timestep interval of 1800 seconds generates a dataset for each user. The worse case is for the application to be synchronized so that the timestep completes at the same time for all users. For row 3 and configuration A, this

results in 2 datasets of size 25GB that need to be migrated over an OC-12c WAN - which takes 12.4 minutes (2*25GB / 67MB/s). Again the numbers assume a homogenous workload of that type for that row.

Type of Run (OC-12 WAN)	Max Num Users	Timestep Calculation Time (sec)	Config A: Vis, Size TS BW User BW Max Lat	Config B: Sub-Domain, Size TS BW User BW Max Lat	Config C: Geometry, Size TS BW User BW Max Lat	Config D: Frame, Size BW Max Lat
Wrkld 1, LLNL: High End 3D	1	150	100GB 667MB/s 3.3GB/s 24.9m	3GB 20MB/s 100MB/s 44.8s	400MB 2.7MB/s 13.3MB/s 6.0s	3MB 30MB/s 30MB/s 0.04s
Wrkld 2, LANL: High End 3D	1	3600	50GB 14MB/s 1.7GB/s 12.4m	1.5GB 0.4MB/s 50MB/s 22.4s	400MB 0.1MB/s 13.3MB/s 6.0s	3MB 30MB/s 30MB/s 0.04s
Wrkld 3, LANL: Full 3D	2	1800	25GB 28MB/s 1.7GB/s 12.4m	758MB 0.8MB/s 50.5MB/s 22.4s	400MB 0.4MB/s 26.7MB/s 11.9s	3MB 60MB/s 60MB/s 0.09s
Wrkld 4, LANL: Scoping 3D, High Res 2D	10	360	5GB 139MB/s 1.7GB/s 12.4m	N/A	72MB 2MB/s 24MB/s 10.7s	3MB 300MB/s 300MB/s 0.4s
Wrkld 5, LANL: Full 2D	50	36	500MB 695MB/s 833MB/s 6.2m	N/A	18MB 25MB/s 30MB/s 13.4s	3MB 1500MB/s 1500MB/s 2.2s
Wrkld 6, LANL: Scoping 2D	100	18	25MB 139MB/s 83.3MB/s 37.4s	N/A	2MB 11MB/s 6.7MB/s 3.0s	3MB 3000MB/s 3000MB/s 4.5s

Table 3: Multi-User Data Set Sizes and Bandwidth (OC-12c WAN Shaded)

The table entries for the sub-domain data set are empty if the visualization data set is not three dimensional (which is the case for the last three workloads). Recall that the sub-domain data set is a surface extracted from a three dimensional visualization data set.

In general, the workloads that use the smaller fraction of the high-end machine generate the most bandwidth requirements. This suggests that if the remote site could handle workloads 4-6 with local resources then the WAN bandwidth requirements would be manageable (in Table 1, the multi-user timestep driven bandwidths less than OC-12c (67MB/s) are in shaded areas).

Some workloads (e.g., workload 1) require very large WAN bandwidths to receive the visualization data sets at the rate they are being created. Also, the size of most data sets are such that even for an OC-12c WAN the latency will range from 6 seconds to 25 minutes, with the exception of the uncompressed 3MB frame data sets (configuration D). If user productivity is unduly impacted by these bandwidth limitations and latencies, then a compromise between the user interaction and remote configuration must be explored.

Conclusion

This analysis suggests that the bandwidth requirements for high-end computing at a distance can be mitigated with reasonable compromises with the user interaction. For example, the user will need to perform some of the data reduction on the distant high-end computer so that the data sets traversing the WAN are of moderate size. This will reduce the user's flexibility in analyzing the resultant data, since the data reduction assumes some a priori knowledge of what aspect of the results are interesting and relevant.

Also, some storage capacity will be necessary to provide for the migration of these data sets from the distant site to the local site. A reasonable complement of computing resources, such as SMPs and visualization servers, are also required at the local sites. This is needed to support the highly interactive user activities such as problem setup, post-processing, and fly-around visualizations.

Technical Information Department • **Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551