A Survey of Co-Processing Systems

Randy Heiland
and
M. Pauline Baker

NCSA
University of Illinois

August 1998

1. Introduction

This report examines systems to support interactive computation, or computational monitoring and steering. Computational monitoring allows the researcher to use computer-generated graphics to visually monitor the progress of a job. Alternatively, interactive computation lets the user access the current state of output data to perform some secondary analysis. Interactive steering extends this functionality such that the user can modify parameters of the simulation while it is running.

Computational monitoring allows the user to access output data as it is produced. This offers the following advantages:

- Monitoring can alert a user to the fact that a run has gone awry and should be stopped without wasting further compute cycles.
- For computations where the data is too large to store or post-process, generating visualizations throughout the life of the run might be the only way to analyze results of the run.

Not all computations are appropriate for computational monitoring and/or steering. Candidates for computational monitoring will typically be an iterative computation (e.g., time-dependent PDE, looping over a spatial domain), and will be parallel and distributed. For those computations that complete very quickly, or that produce only small simulation output, computational monitoring might not be necessary or worth the effort. And for projects that consist of routine production runs, computational monitoring might not add value. However, if the computational output is too large to post-process conveniently, producing visualization during the run might be the only way to get a look at the simulation output. Interactive computation might also be useful to these production-dominated projects in the early stages of the effort, when algorithm development and debugging of the simulation code dominate the effort.

This report surveys existing software packages that offer at least partial solutions to the problem of interactive computation. Our observations are based solely on the documentation provided with each package. A later report will describe our experiences (and those of colleagues) with hands-on use of a subset of these packages.

The primary contribution of this initial report is to provide a synopsis of software packages that might be useful for interactive computation. Our survey will address the following topics:

Availability	Where can I get it? Cost?
App Domain	Is it intended for specific applications?
Platforms	On what platforms will it run?
Dependencies	On what software does it depend?
Functionality	What exactly does it do?

At the end of this report, we provide some relevant links and references.

2. Software Packages

The following packages were selected for this survey. These are the major systems available now that support co-processing.

Freely available	**Restriced availability**
CUMULVS	SCIRun
DICE	AVS
FASTexpeditions	IBM Data Explorer
pV3	
Vis5D	
VisAD	
scivis	

CUMULVS

CUMULVS (Collaborative User Migration User Library for Visualization and Steering) was developed at Oak Ridge National Laboratory. It is described as an "infrastructure library that allows multiple, possibly remote, scientists to monitor and coordinate control over a parallel simulation program". Coordinating control means being able to interactively change values of input parameters, thereby steering the computation.

Essentially, an application program would be modified slightly by calling a few routines in the CUMULVS library. These routines would describe the data distribution and the steerable parameters, and would enable the visualization. A separate "viewer" program is used to request chunks of data to visualize. The CUMULVS software comes with three sample "viewer" programs: a text-only version, an AVS (AVS5) module, and a Tcl/Tk viewer for a particle-based example application. In addition, there is work ongoing at NCSA to provide a viewer based on VTK (www.kitware.com/vtk.tml). From a viewer, one makes requests to the running application for a "frame" of data and the frequency with which to receive frames. (A frame of data would be defined by a range and step size for each coordinate index of the computational domain.) A nice feature is being able to attach and detach a viewer from a running computation.

CUMULVS uses (is dependent on) PVM, but this does not affect the application program. As PVM evolves into Harness (www.epm.ornl.gov/harness), CUMULVS is supposed to evolve also.

The source and documentation is freely available and the User's Guide seems quite adequate for getting started.

DICE

DICE (Distributed Interactive Computing Environment) was developed at the Army Research Laboratory Major Shared Resource Center (ARL MSRC). It is described as a flexible framework for runnning and visualizing computationally intensive codes in a heterogeneous distributed environment. It consists of a distributed shared memory facility, 2D and 3D graphics, GUI tools, and custom interfaces for some existing codes.

At the foundation of DICE is the Network Distributed Global Memory (NDGM) system, also developed at the ARL MSRC. This system, as its name implies, is an implementation of distributed shared memory (DSM), where the distribution can be across a network. The memory buffer that is shared among processes is unstructured. To provide some structure to this buffer, sitting on top of NDGM is the DICE Data Directory (DDD). This uses NCSA's Hierarchical Data Format (HDF). Using the DDD, one can specify that some data is to be stored in a disk file while other data is to be stored in the NDGM buffer.

Another component of DICE is the DICE Visualizer (DV). While this originally consisted solely of custom visualization data filters (e.g., isosurfaces and slice planes), the latest version takes advantage of VTK (the Visualization ToolKit). (Technically, the data filters are not part of the DV).

The DICE GUI sitting on top of all the components uses Tcl/Tk plus some Tcl/Tk extension packages: Tix, BLT, and Dp. The GUI allows interfaces to NDGM, the DDD, the DV, and (optionally) the application code itself.

There seems to be a concerted effort to incorporate DICE into existing, high-demand applications, especially DoD CHSSI applications. Two of these are CTH, a structural mechanics code, and Dzonal, a 3D Navier-Stokes code.

The software is not publicly downloadable. Contact the authors for access.

FASTexpeditions

FASTexpeditions was developed at the Numerical Aerospace Simulation (NAS) Facility at NASA Ames Research Center. As stated in their Overview of FASTexpeditions, its key concept is the following:

> *Instead of sending scientific data over the network as pixels to a movie player, data is sent in a basic form that can be analyzed by FAST, a sophisticated analysis tool, running on the recipient's computer. In addition, scripts for driving FAST are included. These scripts provide a variety of guided expeditions through the data. The recipients can take the guided expeditions, create extensions with their own "what if" analysis, or conduct their own expeditions independently from the guided expeditions. This technique provides a more effective scientific analysis and reduces the load on the network.*

FASTexpeditions uses FAST (Flow Analysis Software Toolkit). FAST offers the following advantages:

- It is available free
- It has a rich set of analysis tools
- It provides the high performance needed for effective interactive analysis.
- It is a complete package. No coding or assembly of modules is required.

Creating the FASTexpeditions is trivial. FAST has a built-in journal feature that automatically records your actions as a script file that can be replayed. A FASTexpedition is essentially just one of these script files. Furthermore, the script files are plain text, so they can be easily edited with any text editor.

It has built-in stereo viewing

It has a large user base already established

FASTexpeditions requires an SGI workstation. (It uses the SGI gl library. Val Watson, the primary contact, indicated that FAST had not been converted to OpenGL and there were no plans to do so.)

pV3

pV3 (parallel Visual3) was developed at MIT and is targeted primarily for CFD codes. It is a client-server system whose ancestor was the Visual3 system. The binaries (in various Unix flavors) for the server and client are publicly downloadable from the pV3 web page, but the source code is not made available. There is, however, source code for example clients (both Fortran and C).

Like CUMULVS, pV3 uses PVM for its communication layer. And like CUMULVS, pV3 is capable of plugging into/unplugging from a simulation and of doing computational steering. Unlike CUMULVS, which has no visualization built in (it uses external packages), pV3 does its own visualization (and is a "closed" package).

There supposedly exists a pV3-Gold which is a version with an "intuitive Motif GUI". However, the link to pV3-Gold from the pV3 home page is apparently dead and attempts at contacting a person responsible for pV3-Gold have so far been unsuccessful.

Vis5D

Vis5D was developed at the University of Wisconsin Space Science and Engineering Center (SSEC). It is a software system for visualizing data produced by numerical weather models and similar sources. Vis5D works on data in the form of a five-dimensional rectangle. That is, the data are real numbers at each point of a "grid" which spans three space dimensions, one time dimension and a dimension for enumerating multiple physical variables. Vis5D works fine for data sets with only one variable or one time step (i.e. no time dynamics). However, your data should have some depth in all three spatial dimensions.

One could argue that this package doesn't belong in this survey, since it seems to visualize only static gridded datasets. However, given that the source code is freely available, it should be possible to enhance it to perform remote visualization and interactive steering. In fact, one enhanced version is *Stream Enabled Vis5D* which can take data as a stream over a network.

Another interesting fact about this package is its support of multi-processing. Quoting from the User's Guide:

> *The SGI version of Vis5D uses multiple CPUs if available to compute graphics in the background thereby increasing Vis5D's speed. On other systems, Vis5D tries to interleave the computation of graphics with user interaction. This results in the user interface being a bit sluggish until all pending graphics computations are completed.*

VisAD

VisAD is a descendant of Vis5D, but with very different genes. It too was developed at the SSEC. VisAD is a pure Java system for interactive and collaborative visualization and analysis of numerical data. It requires Java 3D and Java 1.2 (beta 4, as of this writing). (Vis5D was written in C, as were earlier versions of VisAD). A VisAD application is an event-driven network of: Data, Display, User Interface, and Computational objects.

When asked about the possibility of doing various geometry filters (e.g., isosurfaces, decimation, etc.) in VisAD, Bill Hibbard (the lead author) replied:

> *"Computational geometry is implicit in visualization operations. The current DataRenderers include iso-surfaces for regular and irregular topologies, but no decimation. The irregular Set constructors will apply Delaunay triangulation if necessary. Applications could include extensions of DataRenderer that included decimation."*

Apparently steering is possible in VisAD. Quoting from the Overview chapter of the Developer's Guide:

> *By allowing Data, Computational Cells, Displays and User Interface components to be connected flexibly, VisAD supports computational steering interactions.*

VisAD source code and documentation is freely available. Naturally, one needs to realize that a particular version of Java/Java 3D may not be available on a particular platform/OS.

Scivis

Scivis is a collaborative scientific visualization system available from the Northeast Parallel Architectures Center (NPAC), Syracuse University. Like VisAD, it also uses Java.

Quoting from the "What's New" page:

> *"Scivis has a sluggish performance when the software renders 3D colormapped surface plots. Scivis can be drastically improved if we had hardware acceleration for graphics. We are going to use the Java3D API when it is available in the future releases. So the next release of Scivis will use JDK 1.2 and Java3D API. But since Java3D API is not part of JDK 1.2, two different versions will be released in the future. The difference is the usage between AWT rendering model and Java3D."*

According to the lead author, Scott Klasky, the next version of Scivis is "focusing on very large data sets (1GB), with collaborations of over 10 people. We are designing this to be used with collaborations of scivis3d (Scivis+java3D) and AVS users".

SCIRun

SCIRun is a "scientific programming environment for computational steering" from the University of Utah. It is one project from the Scientific Computing and Imaging (SCI) research group in the Department of Computer Science.

The user interface to SCIRun looks similar to AVS5. A user (graphically) builds a data-flow network by connecting modules together. The modules represent operations on the data.

It seems that SCIRun is not publicly available and will become a commercial product in the future.

AVS

On the AVS web site, we read:

> *For solutions to large-scale problems, AVS is designed to run in a distributed environment. AVS modules can execute on supercomputers and massively parallel machines, helping to bring visual computing to where your data and simulations reside. When transferring data between machines with incompatible data formats, AVS handles all data format conversion issues for you.*

Data Explorer

At the IBM Data Explorer (DX) home page, we read:

> *Applications can also be built on top of portions of the Data Explorer system, so that custom, turnkey applications can be easily developed using DX as the computational and rendering engine.*

The product's FAQ includes the following:

G12*. Can I run DX on a parallel machine?*

There is a separate version of DX, DX SMP, which supports intramodule, shared memory parallelism on SMP machines (Sun SPARCstations and SGI Onyx and Challenge, and IBM RISC/6000 SMP machines). Both DX and DX SMP support intermodule parallelism on workstation clusters and on IBM SP machines.

G13*. Can I build applications on top of DX?*

While Data Explorer is a total visualization environment, you can also build applications which use portions of DX. Thus you can create your own special-purpose turnkey application with a custom look and feel, which uses Data Explorer behind the scenes. For an in-depth description of some of the many ways you can embed Data Explorer functionality within your application see building applications.

3. Summary

In summary, we provide a concise set of tables listing features of the packages described above. For the non-commercial packages, we won't try to explain the licensing issues, e.g., Open Source, copyright limitations, etc., but rather, simply designate it as being available in either Source or Binary (libraries and/or executable).

CUMULVS

Availability	www | Source
App Domain	non-specific
Platforms	
Dependencies	PVM, [AVS]
Functionality	access data from a parallel application; steering; vis via AVS module

DICE

Availability	www | Binary (with permission)
App Domain	non-specific
Platforms	most Unix
Dependencies	Tcl/Tk + Tix + BLT, OpenGL or Mesa, HDF
Functionality	vis, access data from a parallel application

FASTexpeditions

Availability	www | Source
App Domain	CFD
Platforms	SGI
Dependencies	SGI gl lib
Functionality	vis, collaboration

pV3

Availability	www | Binary
App Domain	CFD
Platforms	most Unix
Dependencies	PVM 3.3 or higher, platform-dependent graphics lib
Functionality	parallel vis

Vis5D

Availability	www | Source
App Domain	numerical weather models (rect. grids)
Platforms	most Unix, Linux
Dependencies	
Functionality	vis

VisAD

Availability	www | Source
App Domain	non-specific
Platforms	SGI,IBM,Sun,HP,DEC Alpha,PC/Linux
Dependencies	Java 3D, Java 1.2, Java RMI
Functionality	vis/VR, collaboration, steering

Scivis

Availability	www | Source
App Domain	non-specific
Platforms	??
Dependencies	Java
Functionality	vis, collaboration

SCIRun

Availability	www | To be commercial
App Domain	non-specific
Platforms	Unix (??)
Dependencies	(??)
Functionality	vis, steering, analysis

AVS

Availability	www | Commercial
App Domain	non-specific
Platforms	most Unix,(NT-for Express)
Dependencies	
Functionality	vis

Data Explorer

Availability	www | Commercial
App Domain	non-specific
Platforms	most Unix, NT, W95
Dependencies	
Functionality	vis

4. References

CUMULVS	www.epm.ornl.gov/cs/cumulvs.html
DICE	frontier.arl.mil/clarke/dice.html
pV3	raphael.mit.edu/pv3/pv3.html
FAST	science.nas.nasa.gov/Software/FAST
FASTexpeditions	science.nas.nasa.gov/Software/FAST/FASTexpeditions/home.html
Tecate	www.sdsc.edu/Tecate/tecate.html
DAQV	www.cs.uoregon.edu/~hacks/research/daqv
scivis	kopernik.npac.syr.edu:8888/scivis
MPI	www.mcs.anl.gov/mpi/index.html
PVM	www.epm.ornl.gov/pvm/pvm_home.html
HDF	hdf.ncsa.uiuc.edu
HPVM	www-csag.cs.uiuc.edu/projects/hpvm.html
Charm	charm.cs.uiuc.edu/links.html
EnSight	www.ceintl.com
AVS	www.avs.com
IBM Data Explorer	www.almaden.ibm.com/dx
Division Inc., dVISE	www.division.com/2.sol/a_sw/sol_a.htm
VTK	www.kitware.com/vtk.html
Java products	www.javasoft.com/products

Java 3D API	www.javasoft.com/products/java-media/3D
SCIRun	www.cs.utah.edu/~sci/projects/sci-comp.html