Swat Team Trip Report

Center for Remote Sensing and Environmental Optics
Santa Barbara, CA
Friday, 30 October 1992

Mike Olson (mike_olson@postgres.Berkeley.EDU)

1. Introduction

On Friday, 30 October 1992, the Sequoia 2000 Swat Team visited the Center for Remote Sensing and Environmental Optics (CRSEO) in Santa Barbara. The meeting included swat team members (Jim Frew, Mike Olson, Peter Kochevar, Duane Waliser, and Allison Woodruff) and researchers and staff members at CRSEO (Jim Davidson, Frank Davis, Jeff Dozier, Steve Miley, Walter Rosenthal, Jiancheng (J.C.) Shi, and Dave Siegel).

CRSEO researchers presented their research and discussed areas in which they need more computer support. As was the case at previous meetings, visualization was discussed at some length. At this meeting, two new topics were also discussed with the swat team — computational steering and data lineage. Definitions of these terms and some early ideas for supporting them appear later in this report. Researchers at other sites, who have not yet had a chance to discuss these topics, are encouraged to send their opinions to Mike Olson (mike_olson@postgres.CS.Berkeley.EDU) for inclusion in the final swat team benchmark specification.

2. Overview of CRSEO

The Center for Remote Sensing and Environmental Optics is an independent Organized Research Unit (ORU) on the UCSB campus. Researchers at CRSEO, like those at Scripps, are independent of one another, but sometimes collaborate on projects and in general try to exploit similiarites among their specialties. For example, mathematical modelling techniques developed for one specialty are often directly useful to researchers in other specialties.

3. Data Lineage

At the 1 October meeting of the Sequoia visualization group, the swat team was charged with defining the terms "data lineage" and "computational steering," and with incorporating them into the benchmark specification. In order to get as much input as possible, these topics were covered in the morning meeting at CRSEO, with the entire group.

Keeping track of data lineage means keeping track of metadata that describe data of interest to a researcher. This metadata should be tightly bound to the data it describes, so that it is easy to examine, and so that it follows the data on its path through the system.

Some examples of metadata that must be maintained include:

- A description of the data's format and of any devices used to collect it.
- The processing history of the data, including the code that was executed to transform the data. Note that this description can be arbitrarily fine-grained. For example, it should be possible to determine, for a single pixel of a processed image, what input image pixels contributed to its value, and what algorithm was applied to compute the pixel's value.

- Annotations assigned by humans to data. For example, if a set of field measurements were collected by someone unreliable, the data could be tagged as suspect. Similarly, if an image is oversampled to very high resolution, individual pixel values are untrustworthy, and should be marked as such.

For data stored in POSTGRES, complete time history is stored, but the connections between each step in time are not. In order to maintain the processing history of data, POSTGRES for Sequoia must be extended to maintain these connections automatically. In addition, code that is used to process data should be stored in the database system, so that the correct version of it can be located when processing history is being explored.

Annotation is inherently an ad-hoc operation; some user of data simply wants to attach descriptive text to some or all of it. An appropriate metaphor is the use of Post-It notes. As data is transformed by processing, all of the Post-It notes attached to it should follow it. For example, if a user determines that raw data is entirely unreliable, then any data derived from it will be equally unreliable, and should be so tagged.

An important issue, not resolved at the swat team meeting, is how the enormous amount of metadata for any datum can be managed by a user. Some interface to data lineage must be provided. Designers of Sequoia interface tools should consider this issue as they construct their systems.

4. Computational Steering

Steering a computation means that its course can be changed in real time, interactively, by a user.

Consider a global climate model. As they currently run, these models start from some initial state and run for some number of timesteps. The user cannot interactively change *variables* in the model (*e.g.*, the salinity of a particular ocean cell) or *parameters* of the model (*e.g.*, the solar constant[1]). A model changes the values of its variables, but not of its parameters.

This means that a researcher must re-run the model from scratch, often modifying a computer program to change the parameter or variable of interest. If variables and parameters were under interactive control, then models could be more easily used to explore system behavior.

A number of interesting observations were made at this point. First, viewing models in this way makes them appear very much like spreadsheets. Values in individual cells are linked by systems of equations. Spreadsheet programs permit users to change either the equations (in model parlance, the parameters) or the cell contents (variables). This seems to be a powerful metaphor, and we should consider studying spreadsheets in the context of Sequoia.

Second, Peter Kochevar pointed out that cellular automata are another example of grid cell based computation. In a simple system of automata, a checkerboard is populated with some number of citizens, with no more than one per square. At each time step, new citizens are born on empty squares or old citizens die, depending on the population in the immediate neighborhood. State transitions like these are similar to those in spreadsheets. Cellular automata may be another interesting abstraction to explore for Sequoia.

No actual plan for attacking computational steering was proposed. In the short term — that is, by the January retreat — we do not plan to rewrite existing global climate models as

[1] Climatologists find it useful to change the solar constant, but nevertheless insist on calling it that.

spreadsheets. Instead, project members should continue to think about new modelling paradigms. In addition, new data manipulation software should expose interesting knobs and switches to the user. AVS modules should allow dynamic configuration, to the extent possible.

5. Visualization

CRSEO researchers need the same kinds of visualization support that are required by their colleagues at the other Sequoia sites. The requirements include:

- A tool to convert observational and model data to and from different projections. This includes interpolation of images. As was the case at previous swat team meetings, Jim Frew agreed to oversee the implementation of such a tool. A student at Berkeley has begun working on this. The work will be based on GCTP, a tranform package written in FORTRAN and available for anonymous ftp from postgres.CS.Berkeley.EDU in directory frew/src, file gctp.tar.Z.
- The ability to display objects in a format other than that in which they are stored. As a simple example, double-precision floating point values should be mapped to eight-bit colormap indices for rendering under X11.
- The ability to visualize multi-dimensional data easily, using techniques like volume surface rendering, slicing, and animation over time.

Tools currently in use at CRSEO include

- IPW, a home-brew package that uses Unix pipelines to compose complex visualizations from simple tools;
- IDL, the interpreted language in use at other sites;
- AVS, the visual dataflow system;
- spreadsheets; and
- POSTGRES.

Jim Davidson recently gave a seminar on the similarities and differences between AVS and IPW. Most of the expertise at CRSEO is with IPW, and there is strong interest in turning some IPW modules into AVS modules. The current version of AVS consumes far too much memory, however, so this may not be a good idea in the very short term.

Peter Kochevar recommends two strategies for resolving memory problems under AVS. First, sites should load one or two machines with a lot of memory (128 megabytes makes it possible to use AVS; 64 megabytes is not enough) and use these to do AVS visualization. Second, AVS is working on the next release of their system, and Sequoia has close contacts with the company. We should state explicitly to the developers that memory consumption is a serious problem in the current release.

6. Projects at CRSEO

This section summarizes the projects presented to the swat team during our visit.

6.1. Net Primary Productivity and Wildfire Detection

Frank Davis is using AVHRR satellite images to compute a "vegetation index." AVHRR images are released as bi-weekly composites, meaning that images from two weeks are combined to produce a single cloud-free image. Davis uses two image bands to compute the greenness of a given pixel.

In boreal (northern) forests, the resulting vegetative index maps can be used to identify places that have been burned by wildfires, but this is because wildfires in boreal forests are large and reduce dense forest to ash. In forested regions of the California Sierras, fires are smaller and occur in mixed terrain.

Davis' project is trying to compute the footprint of a wildfire from vegetative index maps for the Sierras. In order to make this easier, actual wildfires are identified by looking for areas saturated in the thermal infrared band. When the fire has burned out or been extinguished, algorithms can concentrate on the region around the fire to identify its borders. Another strategy is to look for wildfires only where they are likely to happen. Crop harvest in agricultural regions looks remarkably similar to wildfire from the satellite images, but wildfires in these areas are improbable.

In order to visualize the resulting wildfire data, Davis would like to animate the vegetative index maps over time. He would like to be able to zoom in and out on the animation interactively.

Davis has a report that will be published in a book by Springer-Verlag describing this work. Mike Olson has agreed to get an electronic copy of this report so that interested parties can get it by FTP from postgres.

6.2. Modelling Penetration of Radiation into the Ocean

Dave Siegel is modelling the penetration of solar radiation into upper layers of the ocean. Most global climate models currently ignore differences in absorption and reflectance at different ocean depths. Siegel is modeling solar radiation penetration using Catherine Gautier's surface radiation flux data, combined with Coastal Zone Color Scanner (CZCS) chlorophyll imagery and ocean mixed layer depth information. The immediate goal of this project is to evaluate the amount of heat that penetrates through the mixed layer and so does not participate in air-sea interaction processes. In many oceanographic regions (in particular, the tropics), these fluxes can be quite large ($O(50\,Wm^{-2})$). The project's ultimate goal is to develop the means to assimilate data from future ocean color imagers, such as the SeaWiFS (Sea Viewing Wide Field of View Sensor), into general circulation models.

Siegel's primary Sequoia-related interest is in the combination of three different kinds of data for his model.

6.3. Turbulence Modelling

Dave Siegel is also modelling and visualizing oceanic turbulence in a triply periodic cube. The goal of this modelling effort is to mechanistically predict vertical mixing coefficients of ocean circulation models. Most available fluid dynamics packages model airflow over wings, and are not useful for this sort of modelling. The model consumes large amounts of Cray time, making it hard to do interactive analysis or computational steering. When Siegel wants to explore details of the model more carefully, he must change the model and restart it on the Cray. Interesting analyses must be run on the supercomputer because the saved model state is not sufficient for local analysis.

Frew notes that if complete model state were stored in the database, then subsets of interest could be exracted easily using selection criteria. This kind of support would make new types of analysis possible; for example, Siegel currently examines only single frames at a time, but if all frames were available, he could look for standing waves in the system.

6.4. Estimating Snow Cover from Satellite Images

Walter Rosenthal is analyzing Thematic Mapper satellite images to compute the location and water content of snow. His approach is to build a simple decision tree that identifies snow and its water content, and then to use that tree to analyze new satellite images. A single image is analyzed numerically using a divide-and-conquer algorithm that runs very slowly, but groups pixels that are most similar into the same subtree. The result is that given a coarse description of a new pixel, the pixel can be located in the correct subtree.

Walter has succeeded in building such a tree from TM satellite data. The next step is to use the tree to analyze other TM images, and to figure out whether the snow location and water content guesses that the tree makes are correct. At the time of the swat team meeting, he had not yet done this, but expects to do so in the very near future.

Walter would like storage and analysis support from Sequoia for this process. Storage is not a problem. Walter uses the S statistical package to construct the decision tree, and S is not integrated into either POSTGRES or the set of Sequoia tools available for use by researchers. We should provide statistical processing using S.

6.5. Estimating Snow Water Content from Radar Images

J.C. Shi is using images from Synthetic Aperture Radar (SAR) to find snow and to compute water content. This collection device allows users of the data to extract any pair of transmitted and received polarizations of the image. Varying the polarization during analysis allows Shi to classify snow, ice, and rock without having topographic data. In addition, snow water content can be computed.

SAR images are in a format similar to Thematic Mapper images, but must be managed differently. The inclusion of polarization information, and the use of the Stokes matrix to allow any intermediate polarization to be extracted, make this kind of data inherently not subsamplable.

7. Miscellaneous

This section covers minor issues that were discussed during the day.

7.1. Sequoia Storage Issues

Sites that want to load data onto Bigfoot right now should use either Jaquith (to store data on the Metrum jukebox) or icopy (to store data under POSTGRES). NFS support for the Inversion file system on top of POSTGRES will be availabe by the end of the year.

Steve Miley would like to understand policy on the use of Bigfoot. For example, should Bigfoot be used in real time by applications, or should data be fetched from Bigfoot and cached locally?

Both Jim Davidson and Steve Miley encouraged the storage group to come up with a strategy for staging data at Sequoia sites that does not require management by the user. For example, every site could be required to buy four gigabytes worth of magnetic disk. Users could ask the storage system to move a file they wanted to local storage from Bigfoot. Space would be automatically allocated on the local magnetic disk cache, and users would be notified (by email?) when their data was available. This space would be automatically garbage-collected. Users would be notified just before their data was removed from the cache.

There was also concern over the backup strategy for data stored on Bigfoot. We have no good ideas on how to back up ten terabytes of storage yet. Researchers with valuable data should keep copies locally (offline is fine) to protect against disasters.

7.2. Memory Problems

As mentioned in the visualization section, AVS uses excessive amounts of memory. Every site should load one or two Decstations or Alphas with 128MBytes of memory and use these for visualizing data under AVS. The project, spearheaded by Peter Kochevar, will work with the AVS developers to find a more elegant solution in the medium term.

7.3. Work on Alphas

Berkeley has an Alpha machine available for use by project members. Everyone on the project should have an account on it already. If this is not the case, contact jon forrest (jon_forrest@postgres.CS.Berkeley.EDU) to get an account. The machine is alpha.CS.Berkeley.EDU, and is currently running a beta version of OSF/1. For serious applications requiring a lot of computational time or memory, please coordinate with Jon Forrest to be sure that you have uncontested access to the machine.

Our agreement with DEC prohibits the publication of benchmark numbers for the machine. Nevertheless, researchers are encouraged to convert their programs to run on the Alpha, and to give us feedback on its performance. Jeff Dozier has agreed to set up and run a radiation model on the machine, and to compare its performance on a Decstation 5000.

7.4. Documentation

Peter Kochevar would like to see a glossary of Earth science terms for the computer science community. Hughes has published a list of acronyms and abbreviations for the EOSDIS project. Frew will try to get the electronic version of this list for augmentation by the project. Earth scientists with such glossaries are strongly encouraged to contribute them.

In addition, the project would like to collect manuscripts and technical reports from the Earth scientists. At present, all of the Sequoia technical reports have been produced by the computer scientists, sometimes with assistance from climatologists. We would like publications describing your work in the field.

8. Summary

The CRSEO trip was as successful as its predecessors. The major new contribution to the swat team's efforts was the definition of data lineage and computational steering. Project members at ESRG and Scripps are encouraged to send their views on these topics to Mike Olson (mike_olson@postgres.CS.Berkeley.EDU) electronically for inclusion in the benchmark specification.