Visualization at Supercomputing Centers: The Tale of Little Big Iron and the Three Skinny Guys

E. Wes Bethel[*], John van Rosendale[†], Dale Southard[‡], Kelly Gaither[§], Hank Childs[*,¶], Eric Brugger[‖], Sean Ahern[**]

January 11, 2011

Abstract

Supercomputing Centers (SC's) are unique resources that aim to enable scientific knowledge discovery through the use of large computational resources, the Big Iron. Design, acquisition, installation, and management of the Big Iron are activities that are carefully planned and monitored. Since these Big Iron systems produce a tsunami of data, it is natural to co-locate visualization and analysis infrastructure as part of the same facility. This infrastructure consists of hardware (Little Iron) and staff (Skinny Guys). Our collective experience suggests that design, acquisition, installation, and management of the Little Iron and Skinny Guys does not receive the same level of treatment as that of the Big Iron.

The main focus of this article is to explore different aspects of planning, designing, fielding, and maintaining the visualization and analysis infrastructure at supercomputing centers. Some of the questions we explore in this article include: "How should the Little Iron be sized to adequately support vis/analysis of data coming off the big iron? What sort of capabilities does it need to have?" Related questions concern the size of visualization support staff: "How big should a visualization program be (number of persons) and what should the staff do?" and "How much of the visualization should be provided as a support service, and how much should applications scientists be expected to do on their own?"

1 Introduction

R. W. Hamming famously asserted: "The purpose of computing is insight, not numbers!" Although few would disagree with this observation, supercomputers are deployed in centers and organizations with a complex web of motivations and goals. Conceivably, each of the hundreds of people involved in these centers sees things somewhat differently. Even if everyone truly believes that the fundamental goal is "insight, not numbers," appropriately sizing the necessary visualization and analysis infrastructure – the Little Iron and the Skinny Guys – takes political will and planning that are often absent. The most famous case of this may be the Earth Simulator, whose funding did not allow for an adequate file system, forcing simulations to be halted for lack of space to store the results. Similar instances of the same lack of balance and focus on the Little Iron needed to glean insight occurs daily at supercomputing centers everywhere.

Supercomputing Centers (SC) are the result of significant investments in resources with the aim of bringing large-scale computational power to bear on challenging scientific problems. Ultimately, the success of these centers is measured by the quality and quantity of science they enable. Additional metrics may include the percent of time the machines are available for users, the number and distribution of jobs and job sizes (some centers focus on support for high-concurrency jobs), relative user satisfaction as measured by surveys, and so forth. Some SCs are "general purpose," meaning they are intended to provide resources for a diverse set of science applications. Others are more focused on specific types of science (climate prediction, weapons simulation, etc.).

From a hardware perspective, SCs typically consist of one or more primary computational platforms, the "Big Iron," which host the large computational simulations. Many SCs include secondary platforms, which we refer to here as "Little Iron." These systems are often used for post-processing activities like visual data exploration and analysis. The staff at SCs typically reflect the mission objectives for the center: personnel to perform operations and system administration, user support, installing and maintaining software, and so forth.

We represent the visualization and analysis efforts at

[*]Lawrence Berkeley National Laboratory
[†]William & Mary
[‡]NVIDIA Corporation
[§]TACC, University of Texas, Austin
[¶]University of California, Davis
[‖]Lawrence Livermore National Laboratory
[**]Oak Ridge National Laboratory

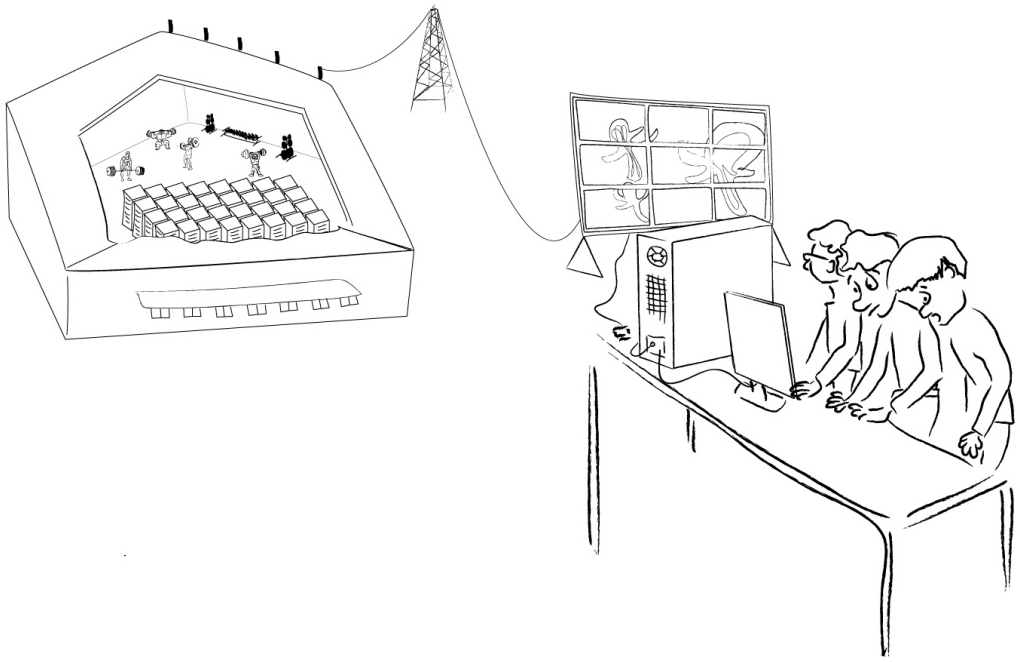

Figure 1: Illustration courtesy Jo Wozniak, Texas Advanced Computer Center, University of Texas, Austin. This image does not appear in the CG&A print version due to space limitations.

the Texas Advanced Computing Center, Oak Ridge National Laboratory, the National Institute for Computational Sciences, Lawrence Berkeley National Laboratory, and Lawrence Livermore National Laboratory. Our experience also covers other laboratories and computing centers. Over the years, we encounter two questions over and over again that are the subject of this article: (1) How big should the Little Iron be at SCs? and (2) What sort of staffing – how many Skinny Guys and their mission – is required to support an effective visual data exploration and analysis program at an SC?

2 Sizing the Little Iron

The question of the appropriate size and makeup isn't unique to Little Iron; each new generation of Big Iron requires scrutiny of this and many other issues. However, Little and Big Iron have substantially different budgets and workload demands.

2.1 Funding Little Iron

Determining how to fund the Little Iron is complicated in and of itself but often falls into one of three broad approaches. We can take a prescriptive approach dictated by the relative size of the compute resource. We can simply size the Little Iron by using the budget that is left over after the compute resource is purchased. Or we can perform a detailed workload assessment and size the Little Iron accordingly. We discuss each of these approaches below and provide some historical perspective.

The *prescriptive approach* is dictated by the computing resource's relative size; it uses a prescribed formula to determine the Little Iron's budget. For example, an SC program might decide to spend 10% of the Big Iron budget to purchase the Little Iron.

The *leftovers approach* simply applies whatever spare change is lying around after paying for everything else. It reflects what happens when SCs carefully plan the Big Iron, then pay for the balance of the infrastructure – file systems, onsite networking, the Little Iron, and so on – out of operational expenses.

The *planned suburbs* approach first performs a carefully planned workload assessment. Then, by hook or crook, it obtains the funding needed to acquire sufficient Little Iron to meet most or all of the anticipated needs.

It is administratively, very difficult to fund Little Iron. Imagine a hypothetical conversation in a bar: Center A staff member: "Our new machine is great. It's ranked X on the Top 500 list and delivers Y petaflops!" Center B staff member: "Sure, but our machine is much better balanced and has a great attached I/O and visualization system. We get a lot more science done!" Even to vis cognoscenti ears this sounds wimpy! "Machoflops" rule!

In general, one would like to choose the "suburbs"

approach when planning for the Little Iron. It makes a lot of real-world sense to first perform a detailed workload assessment, then buy the appropriate amount of "horsepower" to meet those needs. However, funding is often much more complex at large supercomputing centers. The most practical approach is likely to be a blend of all three. Although visualization and analysis are absolutely essential to the scientific understanding that makes a center successful, the resources that enable such infrastructure are typically, but not always, considered late in the process. So the first cut at a budget may likely be from the "leftovers."

These issues played out in complex ways with Lawrence Livermore's purchase of the ASCI (Accelerated Strategic Computing Initiative) White machine at Lawrence Livermore. With far larger simulation datasets than ever seen before, combined with a need to understand weapons physics in 3D for the first time, ASCI faced a massive data-understanding challenge. In response, the VIEWS (Visual Interactive Environment for Weapons Simulation) program was created and was given around 9% of the total ASCI budget at the time for visualization and data understanding infrastructure (hardware and software), and for research and development in this broad area. During acquisition of ASCI White at Lawrence Livermore, the decision was made to spend 10% of the machines' cost on a visualization cluster.

Although this was a consensus decision within ASCI, implementation proved surprisingly challenging. First, although ASCI White was a three-lab resource, there was push back from the other two partner laboratories. They felt that the Little Iron would be useful primarily for interactive graphics at the local institution and wouldn't benefit the remote institutions. This concern was mitigated when staff at those labs learned that emerging visualization tools could effectively support remote, interactive graphics.

The second issue involved the Little Iron's structure. Although initial plans called for a visualization server from a vendor, providing adequate bandwidth from White to an external visualization server would have been challenging. Moreover, the file semantics were so tightly wrapped around White's new file system that there was no plausible way to move files to any other file system. So, the only feasible approach was to allocate 10% of the nodes of White for interactive visualization and data exploration.

The co-location of these nodes on the Big Iron caused a political problem. The visualization nodes were idle when scientists were not doing visualization, which decreased usage statistics. To address this problem, the batch scheduler was modified to allow simulation jobs to run on the visualization nodes, but at lower priority and with the ability to preempt them at any time. (This was, of course, a good choice, since it created additional cycles for the center.)

2.2 Workload Assessment

By "workload assessment," we mean determining the types of applications and their loads that will run Little Iron, along with typical use patterns. The objective is to determine (through empirical data and some guesswork) how big the Little Iron must be to meet most workload needs. Six factors influence workload assessment: the execution model, I/O requirements, in-core versus out-of-core processing, interactive versus batch use, throughput requirements, and GPUs.

Execution model: Two related issues explore the workload's execution characteristics. First, visualization and analysis codes can run in either serial or parallel. Serial codes, which include legacy custom applications as well as numerous commercial products, use a single processor for execution. Tackling larger problems with serial codes will increase per-node memory requirements. In contrast, parallel codes can take advantage of many processors and can often have lower absolute per-node memory requirements while providing greater overall problem size capacity by leveraging a larger, aggregate distributed-memory footprint.

Second, the manner in which users run these jobs can have an impact on the system design. If all user jobs are run interactively, the overall machine utilization might be greater, but with the potential for greater resource contention. In contrast, serializing user jobs through a batch queue will lower contention, but might impact a user's ability to quickly perform analysis.

I/O Requirements: Interactivity depends heavily upon I/O bandwidth; the amount of I/O bandwidth needed for any particular task is in turn a function of the size of data you'll visualize. However, the data's specifics – the data model and format, how it's organized inside a file or files and how bytes are arranged and laid out on storage – are all important. Simulations with low-resolution meshes and many time slices (e.g., many climate modeling codes) suggest a different I/O subsystem design than those with high-resolution meshes and fewer time slices (e.g., turbulence modeling codes). Furthermore, approaches that process the data in a multi-resolution manner can reduce the I/O requirements dramatically, even though the amount of raw data remains large.

In-core vs. out-of-core: Data analysis and visualization applications operate either in-core (reading all data into memory before processing) or out-of-core (reading and processing data one piece at a time). In-core algorithms are much more prevalent, but place large

memory footprint constraints on the Little Iron, because the machine must contain enough RAM to hold the largest possible dataset. Out-of-core algorithms typically require a modest memory footprint, but out-of-core implementations aren't always available or possible for all desired functionality.

Interactive vs. batch use: Utilization of the visualization resources isn't comparable to utilization of the HPC resources at SCs. The visualization resources primarily, but not exclusively, serve as interactive visualization platforms. Because interactive visualization requires the user to be online and interacting with the application, you can't expect the machine utilization to be equivalent to that of applications run in batch mode. For this reason, it is customary to backfill with batch jobs that can make efficient use of the Little Iron resources and minimize dead time.

Throughput requirements: Because visualization is often done interactively, throughput requirements need to be carefully considered as part of a balanced system. One useful metric is "Time To First Image" (TTFI): the time elapsed from requesting a visualization to an image appearing on the screen. This metric, which is relevant for high-throughput scenarios like visualization of time-varying data, depends on a variety of factors, namely, how long it takes to:

- load data,
- apply visualization algorithms,
- render data into images, and
- place the image in front of the user (who could be in a remote location).

An absolute frame rate would be useful for a use scenario where multiple or repeated renderings of a single dataset is the primary objective. In this latter case, meeting some absolute frame rate for interactivity might be important.

GPUs: Depending on the application needs, GPUs may be a valuable part of the Little Iron. Workloads that are heavily dependent upon rendering or compositing may well benefit from having hardware-accelerated frame buffers at their disposal. Additionally, modern GPUs support general purpose programming languages like CUDA and OpenCL, making them useful for accelerating numerically intensive computational jobs. With the HPC community exploring the use of GPUs for HPC applications, the Little Iron can serve a dual role as a testbed for HPC application code development[1]

[1]Author's note: between the time of our submission to CG&A in early Fall 2010 and the January 2011 publication date, the Tianhe-1A system at the National Supercomputing Centre in Shenzhen, China, was announced as being number one on the Top500 list (see `www.top500.org`). The Tianhe-1A system is comprised of NVIDIA GPU processors.

Asking the right questions. How does you determine the workload? Here, the question is, what kind of problems do you expect the future machine to process? Answering this question involves some speculation. One approach some of us have used in the past is to conduct workshops with users to directly ask them what they're doing now and what they expect to do in the future[2].

Because users often don't know what they want, it is important to engineer the questions carefully: How big are your datasets now? How big will they be in 3-5 years? How big are the computational grids (e.g., spatial and temporal resolution) in your data files both now and in the future, number of variables per grid cell, how many time steps, and so forth.

Then, it is important to ask users about their science problems, and how they envision going about solving it. This line of questioning will reveal information about how they expect to use the resources (interactive, batch) as well as provide hints about what kind of visualization they believe they'll need to perform. Collectively, this kind of information helps provide some "error bars" on the anticipated capacity required to meet visualization needs.

2.3 Technology Assessment: What Do You Buy?

SCs tend to be early adopters of the latest and greatest technology. Driven by Moore's law and architectural advances, new technology can offer dramatic increases in performance. Since the priority of SCs is productivity in science and engineering, chasing the newest hardware is natural. And while the newest hardware may be unbalanced or difficult to use, scientists have proven remarkably adept at exploiting new technologies in large-scale computation.

Extrapolating this thought, one might expect that deploying the latest and greatest technology in the Little Iron is the right approach as well. Here, the situation is, however, quite different – new technology has less impact than you would expect. The reason is that, unlike the large scale simulations running on the Big Iron, the challenge in visualization and interactive data exploration is throughput rather than total flops. Although major technology shifts can indeed provide dramatically greater capabilities in the Little Iron, it is only after these shifts percolate through the software stack that one sees their real benefits in visualization and data exploration.

[2]B. Hamann, E. W. Bethel, H. Simon, and J. Meza. The NERSC Visualization Greenbook: Future Visualization Needs of the DOE Computational Science Community Hosted at NERSC. The International Journal of High Performance Computing Applications, 17(2):97–124, 2002.

Typically, visualization systems consumer data generated elsewhere. Thus, great care must be put into throughput and systems balance. Unlike the Big Iron, where clever scientists and engineers can sometimes overcome systems imbalances by judicious choices of algorithms, there are few quick fixes for systems imbalances in the Little Iron. Thus, spending disproportionately on memory without increasing bandwidth to the file store would simply result in a system with a high TTFI that struggles to provide interactivity. Similar interdependencies exist between CPUs, GPUs, memory size, storage, and networking. Technologies such as GPUs, FPGAs, and hardware compositors hold great promise for accelerating graphics, but it is only after such technologies are integrated into a balanced system that they become generally useful.

2.4 Does Separate Little Iron Have a Future?

The hardware landscape changes constantly, and we must ask if current and past approaches are scalable and tractable. We must evaluate whether these approaches need to evolve to reflect changes in workload needs and machine architectures.

As we move from petascale computing to exascale, the forces weighing against fielding separate visualization platforms seem to be growing in strength. Unless the exascale system uses a hardware design that is fundamentally incompatible with visualization (e.g., tiny nodes based on embedded CPUs and very small amounts of memory), it may be more reasonable to figure out how to use a slice of the exascale supercomputer as an interactive visualization resources cluster rather than to lobby for a separate Little Iron machine. Using a slice of the supercomputer as an interactive visualization resource has a number of intrinsic advantages:

- Little Irons are becoming more expensive to field because they need a proportional amount of memory to the Big Iron, and memory is not getting cheaper as quickly as FLOPs.
- The slice that is the Little Iron contributes to the aggregate flop-count of the Big Iron rather than being seen as a "parasitic" expense.
- Obtaining adequate I/O bandwidth to the Little Iron slice is surely easier as it is on the same Big Iron communication fabric.
- The Little Iron slice will automatically be of the same vintage as the Big Iron, and will age (become obsolete) at the same rate as the rest of the Big Iron.
- A Little Iron slice will usually be much better suited for visualization tasks that are closely coupled with running simulation codes, tasks like in-situ visualization, computational steering, and interactive debugging.

Compelling arguments might exist for a separate Little Iron resource to augment exascale-class machines. One argument concerns the availability and support for software applications. Commercial software vendors also have limited budgets and staff and will focus their energies on supporting platforms where it makes economic sense. They likely won't be eager to port their application to a platform where there are only one or two systems in existence. Related, relying on the "open source community" to perform ports of packages to these unique systems may not be as fruitful as expected: in the open source world, the software itself may be free but the people that do the work must still be compensated for their effort. Another argument relates to how SCs desire to maintain very high levels of Big Iron utilization, but interactive visualization tends to produce "bursty" loads on the resource.

Overall, the issue of whether or not there will continue to be separate Little Iron resources is very much open: the answer is it depends upon a number of factors. Each SC must evaluate for itself which route to take given their budget for hardware, software and staff; Big Iron hardware/software choices, operational policies, evaluation metrics, and science objectives.

3 Sizing the Staff: How Many Skinny Guys?

One fundamental issue facing SCs that have a visualization effort is determining how many visualization staff/experts are needed to meet mission objectives. The answer to this question stems from the mission of the center: is it a "lights-out" operation that provides only cycles? Or, does the SC envision providing expert help to make effective use of the its resources?

In a recent advisory program review of an emerging visualization program that is part of a well-known SC, the review panel was given the a list of questions, one of which was the following: "Do we have sufficient staffing resources?" The program wanted to provide state-of-the-art visual data exploration and analysis resources to its science users, but had only one full-time staff person. The reviewer's response was "All these objectives are worthwhile, but you have only *one skinny guy* to do all this work. You need more staff if you want to accomplish all these objectives."

In our experience, the following are factors that could be taken into account when determining appropriate staffing levels for a visualization effort at a SC.

Ongoing care and feeding of infrastructure: This activity includes acquiring, installing, and maintaining both hardware and software. Hardware includes computational platforms, networking, file systems, and so forth. Software includes visualization applications, configuration for use on your particular parallel infrastructure, setting up and maintaining user documentation on the web, possibly doing live or pre-recorded tutorials, and so forth.

Typically, an SC has a group responsible for system administration of platforms and these folks take on the added responsibility for administering the Little Iron. In some cases, there is a group at the SC whose responsibility includes installing software on the SC platforms (e.g., compilers, profiling tools, numerical libraries, and applications). Often, those staff can also take on the relatively small incremental amount of work for installing some visualization software. On the other hand, some visualization software, particularly parallel applications with client/server modes of operation, can be tricky to install and may require a complete recompile from source. In our experience, these activities can consume a non-trivial amount of time.

Consulting, helping users: In a "lights out" operation, the only real user support you can expect is help with forgotten passwords and similar operational issues. In more advanced facilities, working with users one-on-one is where there is real added value. One activity common in many "full service" SCs is help for users in scaling their codes. This activity will include diagnosis of poor performance, consulting with using advanced numerical solver libraries, and so forth. For visualization, there is a parallel theme: how to use visualization tools to create images, movies, and to perform problem-specific work. A lights-out operation tacitly assumes that the software works perfectly. However, if the software breaks or must be adapted or optimized for a particular machine or user problem, much more time will be required than is typical for a lights-out operation.

Applied research and development: To deal with software that doesn't work perfectly or is missing a small but crucial feature, some full service shops have staff who perform applied R&D. Activities here could include extending existing tools, or developing new, custom ones.

In terms of extending applications, a good example for visualization is the ongoing issue of data models and formats. Most contemporary visualization applications can load one of several different formats, often none of these are compatible with the user's data. In some cases, a straightforward data conversion operation from the user's format to a supported formats will work. In other cases, it is necessary to implement a data loader for the visualization tools. Examples for when this type of work is necessary is when data is so large it is impractical to make copies, or the data is of such a form that there is no suitable one-to-one conversion possible (e.g., multigrid problems in climate like cube-sphere or multiple geodesic grids).

How Many Skinny Guys? There is no hard-and-fast rule regarding how many Skinny Guys a visualization operation needs. We've found that in-depth support requires about one Skinny Guy for every three or four in-depth projects per year. An entire Skinny Guy or two's worth of effort can go into installing complicated visualization software, doing user documentation and training, and taking care of ongoing operational concerns, depending up the number of applications and users.

Also, in our experience, these in-depth Skinny Guys produce some of the most tangible impact for the SCs. The SC's role is to enable scientific progress, and the most visible result of scientific progress is insight gleaned through visualization. In many cases, a single image or movie is the result of many hours worth of effort from a small squadron of Skinny Guys working closely with scientific stakeholders over a long period of time.

4 Conclusions and Future Thoughts

The primary role of the SC is to enable scientific discovery through use of large-scale computational resources. Historically, SCs are comprised of large, Big Iron platforms that generate vast amounts of data; visualization and analysis are performed as post-processing activities using Little Iron. Big Iron designs and budgets are carefully built around anticipated workload, which primarily consists of computationally intensive codes. In contrast, visualization and analysis workloads have different characteristics: they are data intensive and thus need proportionally more I/O and memory. The difference in workload, the demands of showing good utilization rates, and other factors, have led to a bifurcated approach to providing a suitable resource pool to the scientific community. One drawback that is common across all our experiences is that while Big Iron facilities are carefully planned and funded, there is wide variance in dealing with the issues of planning and funding the Little Iron.

Many current trends not readily foreseeable a decade ago suggest we may need to modify our approach to these Little Iron issues. The evolution beyond petascale suggests a "perfect storm" of technical challenges. These include a huge problem with I/O and the fact that the cost of a suitably sized Little Iron system may become prohibitively expensive should we continue along the current trajectory. Some feel that using a slice of the Big Iron for interactive visual data exploration and

analysis may be a fruitful approach for a number of good reasons. This approach, however, requires careful attention to a number of design and operational issues that are typically outside the scope of usual Big Iron considerations.

The success of SCs depend upon the amount and quality of science they generate. As visual data exploration and analysis are an integral part of the scientific process, we feel that adequate provision for suitable infrastructure is an essential ingredient to the future success of these centers. Sizing and provisioning hardware is relatively straightforward compared to sizing and provisioning the visualization experts who enable the scientific discoveries. The most visible result of science – the discoveries – are the product of collaborative work of computational and visualization scientists. The Skinny Guys do more than install software and make images for users. They consult with user teams to help them solve difficult issues that include data models/formats, they extend visualization systems to have new capabilities to meet specific science needs. Ultimately, it is these Skinny Guys that enable the visual and analysis scientific data understanding by which these centers are judged.

Acknowledgment

This work was supported by the Director, Office of Science, Office and Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET); was supported in part by the National Science Foundation grants OCI-0906379 and OCI-0751397 and The University of Texas at Austin Texas Advanced Computing Center; and was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, and of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Legal Disclaimer