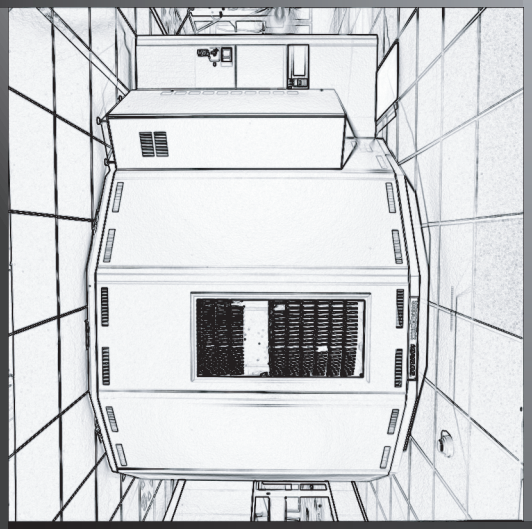
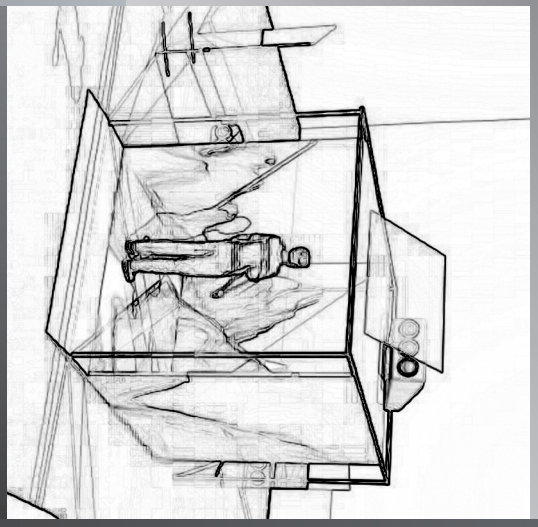
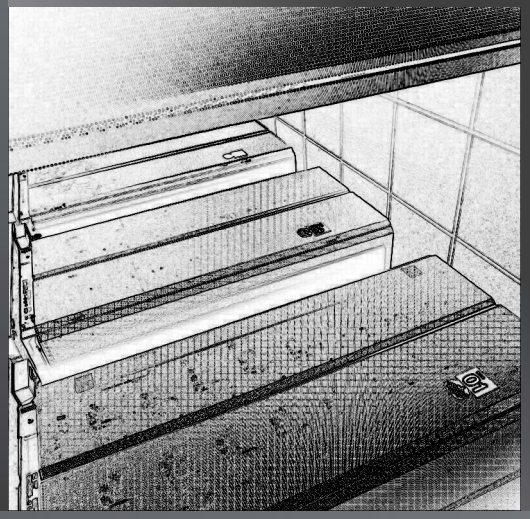
The Least Every Researcher Needs To Know







UNIVERSITY INFORMATION TECHNOLOGY SERVICES

Advanced IT Facilities

University Information Technology Services' Advanced IT Facilities

The least every researcher needs to know

Preface

The Indiana Genomics Initiative (INGEN), Shared University Research grants from IBM, and Major Research Initiative grants from the National Science Foundation have added significant resources to Indiana University's already prodigious advanced information technology systems and services. These new and expanded facilities have been created in order to make possible the new calculations, analysis of massive amounts of data, and visualization of data that will enable IU researchers to achieve new breakthroughs in the humanities, arts and sciences. The purpose of this document is to introduce IU's advanced IT facilities to researchers at IU, clarify what these facilities make possible, and discuss how to use them.

It may seem incongruous to find a roughly 50-page document carrying a title of "University Information Technology Services' Advanced IT Facilities: The least every researcher needs to know." However, the resources described here are complex and varied, and are among the most advanced in the world. Indeed, as of this writing the IBM Research SP is one of the largest university-owned supercomputers in the US.

This document is designed to be read as a printed document, and designed to permit anyone at all familiar with computers and the Internet to start at the beginning, get a general overview of UITS' advanced IT facilities and what they offer, and then read the detailed portions of the document that are of interest. In many cases, examples are provided, as well as directions on how to download sample files. And in some cases there is information that one is best off really not learning – for example the process of logging into IU's IBM supercomputer the first time involves setup steps that should be followed, keystroke by keystroke, from the directions presented herein, and then promptly forgotten.

This document is intended to be a starting point, not a comprehensive guide. As such it should get any reader off to a good start, but then point the reader in the direction of consulting staff and online resources that will permit the reader to get additional help and information as needed.

Most of all, this document is provided for the convenience of researchers, who may peruse this information at their leisure. Our hope and expectation is that consultants in UITS will provide extensive help and programming assistance to IU researchers who wish to make use of these excellent IT facilities.

I. Introduction

There are fundamentally two reasons for using the type of advanced IT resources provided by UITS:

- To do things you can already do, but do them much faster or much better
- To do things you cannot do now.

Routine computing problems are also important, but a document describing all of UITS' baseline services would be long and *very* boring. A description of baseline research services available through the Research and Academic Computing Division of UITS is available online at:

http://www.indiana.edu/~uits/rac/

For information not contained in these Web pages, send e-mail to *rac@indiana.edu.*

One important thing to know about UITS baseline services in support of research and education is that they are almost exclusively offered without any direct charge to the user, as long as the following conditions are met:

- The user is not part of an entity recognized by the University administration as an Auxiliary Service
- The purpose is research or education

The purpose of this document is to help IU researchers and support staff learn about, get accounts on, and get started with the advanced IT facilities available through UITS. The title of this document stems from the goal of getting you going using these facilities with just the critically necessary information – nothing extraneous.

Terminology: Bits, Bytes, and FLOPS

Following the lead of the sciences generally, computing has over time developed its own private dialect of the English language. A comparison of data sizes may be of some use:

- **Bit** – the fundamental (and smallest) unit of data representation within a computer. A bit is always either a 0 or a 1.
- **Byte** – A byte is the smallest unit of data representation usually used by a computer. A byte consists of 8 bits and represents roughly one character.
- **Megabyte (MB)** – a million bytes, or about that same number of characters. There are about 600 MBs of data on a typical CD.
- **Gigabyte (GB)** – a billion characters.
- **Terabyte (TB)** – a trillion characters. All of the printed text in the IUPUI library is about 1TB.

- **GigaFLOPS** – one billion FLOPS (Floating Point Operating Per Second). If a computer performed one floating point operation per clock tick, then the FLOPS count would always be the same as the processor chip's clock rate. In practice this isn't so. The theoretical peak FLOPS rate can be faster for processors that perform more than one mathematical operation per clock tick. In practice computers never achieve their peak theoretical FLOPS count.
- **TeraFLOPS** – one trillion FLOPS. Indiana University's IBM SP supercomputer has a peak theoretical capacity of 1.005 TeraFLOPS.

II. Advanced IT services offered by UITS

As with UITS research computing services generally, advanced IT services are available to IU researchers (for research purposes) without charge.

UITS' advanced IT services are focused on three main areas:

- **Supercomputing.** By using IU's supercomputing facilities, IU researchers can drastically accelerate computing work being done now, and perform massive new computations that are not otherwise possible.
- **Data Storage and management.** IU's massive data storage facilities make possible the secure, reliable storage of massive amounts of data. Anyone who uses more data than fits conveniently on a single CD can benefit from using this system. And anyone who has ever said, "I'd like to pursue this research idea, but I don't have the facilities to store the data" will most likely find storage of data no longer a limiting factor. In addition, UITS supports use of research databases and is developing facilities for publishing research data via the Web.
- **Visualization.** As it becomes increasingly necessary to understand large amounts of data in order to make advances in research, the ability to conveniently and effectively visualize that data becomes ever more important. Visualization tools allow researchers to analyze their data in whole and in detail, and enable them to rapidly spot trends, identify anomalies, and recognize relationships. Furthermore, many research subjects are fundamentally 3D objects, so understanding and insight may often be more rapidly gained by using true 3D views of structures rather than 2D projections or cross-sections. The staff of the UITS Advanced Visualization Lab (AVL) are available to help you analyze your data with the latest visualization techniques and technologies.

An ongoing task for UITS is to make these advanced facilities more easily accessible and simpler to use. As one School of Medicine researcher put it, "I want all of these fancy facilities to appear simply to be an extension of my desktop computer." We are working specifically on this task, but it is an ongoing task. Meanwhile, it can be a bit of work to learn how to use the advanced IT facilities UITS now makes available.

III. An introduction to Supercomputers

Supercomputers (also called high performance computers) make it possible to run existing software applications more quickly and to perform computations that would otherwise take unreasonably long. As of this writing (Fall 2002), we are working with one School of Medicine researcher who uses a data analysis program that typically takes a week to analyze a data set on a computer in his lab. We already have this program down to completing in less than a day, and hope to get the time to completion down to about an hour.

There are three ways that supercomputers enhance the speed with which programs are executed: better uniprocessor performance, trivially parallel processing, and parallel programming.

Better uniprocessor performance

Better uniprocessor performance means that a computer program is running on a single processor - just as it might on your personal computer or a server in your lab or department. However, programs will often run faster on one of IU's supercomputers than on such other computers for three reasons:

- Because of the combination of chip speed, caching and other optimizations in the chip, as well as optimizations of software, the performance of supercomputers processing numerically intensive tasks is generally better than typical current desktop systems.
- The supercomputer has more memory than most computers, so it can hold more data in memory at a given time. While your personal computer might have to write pieces of intermediate calculations out to a hard disk because of memory limitations, a supercomputer is much more likely to be able to hold everything in memory.
- A supercomputer has better hard disk performance than a desktop system. There are a number of features engineered into IU's supercomputers that make data transfer to and from hard disk faster than most microcomputers or small servers.

All of these factors make supercomputers faster than microcomputers or departmental servers for many purposes, but unless your workstation or departmental server runs the Unix operating system, the environment provided by supercomputers will differ from what you are used to. However, for many commercial programs such as SPSS, SAS, and Matlab that run on Microsoft or Apple operating systems, there are also versions that run under Unix. In addition, if you have programs that you or a colleague has written for the Unix operating system, altering ("porting") this program to run on IU's supercomputers is typically very easy (and something we can do for you). Programs written for DOS are also easy to port. Programs written directly for Microsoft or Apple operating systems may be harder to port, but this is also the least common circumstance in practice.

So, the simplest method of speeding up computer programs is to take a program that you are currently running on a personal or a departmental computer, and with the help of UITS staff, convert it to run on one of IU's supercomputers. This sort of conversion might typically take days or at most a few weeks for UITS staff to accomplish.

Trivially parallel processing

"Trivially parallel processing" is a computer science term given to a type of parallel processing that can produce high nontrivial speedups in the completion of your research. "Perfectly parallel processing" might be a better term. Suppose you have a program that runs on a single processor, but you have many different data sets to analyze. Thanks to the many processors of IU's supercomputers, you can actually launch the same program dozens of times, each time analyzing a different data set. This can be done by hand or by automating the process through computer programs called scripts.

This is a very straightforward way to achieve tremendous speedups in the completion of your data analysis. UITS staff can help you implement the scripts necessary to do this sort of parallel processing in a matter of days or a very small number of weeks.

Parallel programming

Parallel programming is the most difficult way to achieve speed enhancements through use of supercomputers, but is also the technique that holds the greatest promise to make possible research that currently is not. This technique involves writing a computer program that takes a single task and splits it up among multiple processors that work on a single problem simultaneously. Unfortunately, it is far from simple to double the number of processors working on a task at once and cut the time required to execute the program in half. In fact, doing this is the theoretical limit of what parallel programming can do and this limit is rarely achieved. Still, there are certain tasks that can be made either possible or practical only in this fashion. This is complicated work, and it can take weeks to months to convert a program from running on a single processor to running in parallel on several processors at once. Still, UITS has worked with IU researchers to parallelize programs and as a result make it possible to do work that would have otherwise taken years to process on a computer and instead do that work in a small number of months. Parallel programming is not the simplest method, and should be the last step in trying to speed up your research computations – but it may also be the most *important* step in speeding up your applications!

As of this writing, IU has three supercomputers – an IBM SP, a Sun E10000 and a large Linux cluster. Which platform or platforms you choose to use will be a matter of your programs' requirements in terms of operating system, software packages, compute capacity and available memory, as well as your own personal preferences. More information on using these systems is included in the "Technical Details" section of this document.

IV. Data Storage and Management

Massive data storage

A typical researcher today is used to dealing with all sorts of computer data – data from field research, data from instruments, from surveys, data generated by supercomputers, etc. A lot of such data ultimately end up on the hard disk of a desktop PC or on a departmental server somewhere. The discipline, time, and cost of hardware required to back up these data means that often, there are no backups available when a hard disk is lost. Consider also the following, typical situations:

- You have an instrument in your lab which wants to produce high-resolution or high-quality data at a rate or a volume that your instrument's disk cache cannot handle;
- You wish that you could access your data securely and transparently, from anywhere;
- You are unable to take on a research project because you do not have access to hundreds of gigabytes or terabytes of storage;
- You need a place to store your data during a break from your research.

If any of these sound familiar, IU's massive data storage facilities are here to help. Based on how you want data to be stored and accessed, two main data storage systems are available at IU for use by researchers.

The Common File System (CFS)

If you are using a PC running Windows or a Mac and want to store your daily work files or other, frequently accessed data on a system which is backed up regularly (by someone else!) and which allows easy, Web-based access, the CFS is for you. Files stored on the CFS remain on disk forever (in contrast to the MDSS service described below) and are thus accessible quickly and securely over the network.

The Massive Data Storage System (MDSS)

Delivered using High Performance Storage System (HPSS) software, the Massive Data Storage System is intended for users with projects that need large-scale, archival, near-line storage. Since HPSS works best with large files, optimal applications will store data in files that are typically larger than 50MB. Some examples include:

- Temporary or archival storage for experiments which generate massive amounts of data aggregated in large files
- Simulation codes generating very large restart files, which need to be archived temporarily

- Computed instrumentation models, which need to be stored for a long time for calibration and data analysis

In general, anything that's too large to fit onto the available on-line storage, and anything that is no longer needed on-line, but needs to be archived off-line or near-line for possible future review is appropriate for the MDSS.

More information on using these systems is included in the "Technical details about IU's advanced IT facilities" section of this document.

Databases and access to biomedical databases

IU's Sun E10000 and the IBM Research SP include components specifically configured to support research databases. If you have data that is already stored in a relational database, or you have data that you would like to be able to store, query, and retrieve easily, UITS can set up a database on these systems. These databases can be set up so that only those people you specify can access them. By using these central systems, you can have the advantage of extremely good database performance, database administration provided by UITS, and automated backup of your database.

We are also able to keep mirror copies of some databases here at IU. If you make extensive use of a public database and would like to have a locally accessible copy, we can provide that (so long as the database permits mirroring). We are also able to provide access to licensed data. For example, the GCG database is accessible via a Web-based front end from a UITS system (in this case it happens to be hosted on UITS's Sun E10000, although this is transparent to people who use this database). UITS operates a Teiresias server for discovering patterns within sequence data, a portal to IBM's BioDictionary database of recurrent patterns of amino acids in 17 genomes, and a portal to IBM's collection of 76 genomes that have been annotated from the BioDictionary database.

More information is available in the "IBM bioinformatics tools on the SP" section of this document.

V. Visualization

Visualization technologies and techniques are perhaps the most specialized of the advanced IT capabilities supported UITS, so it is difficult to provide information and instructions that are of general use or interest. The most effective way to get started with visualization (or to enhance your current work with visualization) is to arrange a consultation with our staff (*avl@iu.edu*).

To illustrate some possible uses of visualization in your research, this section describes some general application areas and types of visualization systems. The section "How to use advanced visualization facilities" details specific visualization services and facilities

offered to IU researchers. Be sure to visit our Web site for the latest information and updates on projects and technologies:

http://www.avl.iu.edu/

Types of Visualization Systems

In addition to understanding your analysis needs and how to best represent your data visually, it is important for us to consider how visualization tasks factor into your scientific workflow. Your needs may fall into one or more of the following categories:

- **Desktop Visualization** – In the most common scenario, you simply use a workstation on your desktop or in your lab to run visualization software, be it commercial, open source, or custom-coded. Your data may reside on the local file system or a mounted file server, and all computation and visual display occurs on the local system.
- **Distributed Visualization** – In some instances, your visualization and analysis work may require the use of special resources available via the network, such as parallel computation from a supercomputer, retrieval of large amounts of data from the mass store system, or access to remote hardware or instruments. Working in conjunction with other groups within UITS, our visualization staff can help design a system that maximizes your efficiency with these distributed resources.
- **Advanced Hardware and Displays** – In other instances, your visualization task may benefit from special purpose hardware such as large-format stereoscopic displays, very high-resolution displays, haptic (force feedback) devices, or computing systems with more memory or processing power than your local system. In such cases, you are invited to use the resources of the Advanced Visualization Laboratory (AVL) in SL 239 on the IUPUI campus and in Lindley Hall 135 on the IUB campus. Should any of these technologies prove to be highly beneficial to your work, we can work with you to identify budgetary and technical requirements needed to acquire and deploy that technology in your lab.
- **Tele-Collaborative Systems** – Finally, you may find it necessary or desirable to collaborate with colleagues who are geographically remote, be they across the campus, across the state, or around the world. Our staff has experience with a full range of tele-collaboration tools, including video conferencing, application sharing, shared collaborative workspaces, and remote hardware access. We can help you utilize these technologies, either in central facilities or directly from your lab or office.

Of course, your needs are likely to change over time, so part of our expertise is used to help you select visualization tools that have the potential to grow with your needs.

VI. High Performance Networking

High performance digital networks and distributed software systems are transforming the way we work, communicate, learn, retrieve and store information, and conduct research. Indiana University, through a number of major developments at the state, national, and international level, has been able to provide the IU community with unprecedented access to high bandwidth networks. IU has achieved a position of prominence in advanced networking through a number of efforts, such as participation in the Internet2 Abilene network.

What is Internet2?

Internet2 is a non-profit consortium led by over 180 US universities working in partnership with over 60 companies. The primary goal of Internet2 is to develop advanced computer network applications such as remote scientific instrument control, high definition video conferencing, and other forms of collaboration supporting instruction and research. Internet2 is a project of the University Corporation for Advanced Internet Development (UCAID). For more information, see:

http://www.internet2.edu/

What is Abilene?

Abilene is an advanced backbone network that connects regional network aggregation points, called GigaPoPs. Abilene is a project of the University Corporation for Advanced Internet Development (UCAID); the Abilene network was developed in partnership with Qwest Communications, Nortel (Northern Telecom), and Cisco Systems. UCAID administers the network in support of all its members, including participants in the Internet2 project.

Abilene complements existing research networks already being used by UCAID member researchers and educators. A primary goal of the Abilene project is to support and encourage the development of advanced applications by UCAID university members and, in particular, to support Internet2.

Indiana University is one of a select number of universities that participated in Abilene's Launch Group. IU runs the Abilene Network Operations Center.

How does Abilene relate to Internet2?

Abilene's advanced capabilities help Internet2 members develop and deploy new applications more quickly and more broadly. Independent of Abilene, Internet2 working groups are tackling networking development issues such as Quality of Service, multicast, and IPv6. Abilene facilitates this work in support of Internet2 and its mission.

How does a user at Indiana University use the Abilene Network?

The mystery of "how to use Abilene" is handled by rules embedded in the network call "routes". Connections between two (or more) computers on an Internet are created according to the routing rules. Our rules are designed to favor sending traffic across the Abilene backbone IF to destination computer is located at an Internet2, Abilene connected institution.

Thus, the user is not burdened with knowing any special commands or network expertise. Procedures like FTP, Remote-Login, and SSH will automatically go across Abilene if the end-points of the connection are all Abilene participants.

Videoconferencing

Indiana University offers several videoconferencing services, including administrative group videoconferencing, desktop videoconferencing, and streaming media. UITS Digital Media Network Services (DMNS) strongly recommends the Polycom ViaVideo appliance for university-based desktop video conferencing. DMNS has evaluated a variety of products and has found that ViaVideo provides excellent video and audio quality, integrates with the university room-based video conferencing systems, and works well with systems used at other institutions. For more information, see:

http://www.indiana.edu/~video/

VII. Getting started using UITS' advanced IT facilities

There are several steps in the process of starting to take advantage of the advanced IT facilities made available by UITS. In the most likely order, they are as follows:

- Ask yourself "what am I doing now with computers that I would like to do faster, more conveniently, or on a larger scale?"
- Ask yourself "what kind of research questions would I tackle if computing resources were not a limiting factor?"
- Read this document to get a better idea as to the kinds of research computing possibilities possible through use of UITS' advanced IT facilities.
- If it is clear from this document what IT facilities are of value to you, get accounts on these facilities.
- Contact the High Performance Computing support team by phone or e-mail (*hpc@indiana.edu* or 812-855-2632) to discuss what you want to do, and let us help you get going.

(The last two items are of course interchangeable based on your preferences.)

How to request accounts

To request an account on the IBM Research SP, the Sun E10000, and/or the AVIDD Linux Cluster, visit

http://www.indiana.edu/~rats/application.shtml

For information on requesting a CFS or MDSS account, see the section "How to use the Common File System or Massive Data Storage System."

As mentioned earlier, these services are available to you without charge so long as the purposes are for research or education. We ask only three things:

- That you provide a very brief (less than 200 words) description of your research activities.
- That you agree to respond to periodic requests for citations so that we can document the impact IU's advanced IT facilities is having on research at IU.
- That you agree to acknowledge use of IU's advanced IT facilities in the methods or acknowledgements sections of your publications.

Basics of the Unix operating system

The Unix operating system is a powerful environment designed by and for computer programmers. Beginners and casual users often find the jargon-filled help system frustrating and the lack of icons and menus unfriendly. But before you get discouraged, remember that many users spend little time at the Unix command line, working instead within applications. This document will show you how to get past the Unix command line as painlessly as possible, and into the applications that you need.

Typographic conventions used in this document

When you are instructed to type some text, do so and then press RETURN or ENTER (depending on your keyboard).

`Text that appears on the screen is in Courier font.`

Text that you are to type verbatim is boldfaced.

Text that you can change, such as a filename, is italicized.

Unix is case sensitive, meaning that when you type a command, you must use uppercase or lowercase letters as they appear in this document. (Most commands in Unix are lowercase.)

The Unix system puts a prompt on the screen to let you know that it is waiting for your commands. Prompts may vary (some common ones are "$," "%," and ">"). In this document the system prompt is represented as $.

Below is an example of the command you would enter to get help:

```
$ man help
```

To see the online manual for a particular Unix command, at the Unix $ prompt, type **man** and the name of the command, and then press the RETURN or ENTER key. In this example you would see the manual page for the help command, but the italics indicate that you can replace ***help*** with another manual subject.

When you are instructed to press a key, press that key alone and do not press RETURN.

Combination keystrokes are joined with a slash. For example, CTRL/z means hold down the CTRL key while you press z.

Common error messages

Like the manual pages, Unix error messages are helpful to programmers but can be frustrating for beginners. Here are some common Unix commands (in bold) and the error messages that may result from them.

```
$ mv
Usage: mv [-i | -f]  [--] src target
   or: mv [-i | -f]  [--] src1 ... srcN directory
```

A usage error is one of the most common (and most opaque) error messages. It often occurs because you need to supply one or more filenames or directory names. In the example above, when using `mv`, you must also give the old (`src`) and new (`target`) names of the file.

```
$ more FitchCode
FitchCode is a directory
```

The **more** command works only on files. You will get this error if you try to use **more** on a directory, such as *FitchCode*.

```
$ spss
spss: Command not found.
```

In this example, `Command not found` indicates that SPSS is not installed on the node you are logged into. (On the IBM Research SP, if you are logged onto aries07, you cannot start an SPSS session. You must be logged onto aries05 for SPSS.)

```
$LS
LS: Command not found.
```

Remember that Unix is case sensitive. If you have your caps lock set and want to list the contents of your working directory, you must turn off the caps lock first.

What to do when confusing things happen

If the DELETE and BACKSPACE keys don't seem to work correctly, try using

CTRL/h instead. You can sometimes fix these keys for the rest of your login session by typing one of the following lines at the $ prompt:

```
stty sane
stty erase ^h
stty erase ^\?
```

If the commands you type do not show on the screen, you may have accidentally frozen your screen by pressing **CTRL/s**. Unfreeze it by pressing **CTRL/q**.

If you accidentally type something you didn't mean, you may find yourself in a program that you don't recognize. Don't panic! Here, in order, are some common exit commands you might try entering:

q, Q, quit, exit, stop, bye, logout

CTRL/c

CTRL/z (When you try to log out, you will get the message: There are `stopped jobs.` Simply type **logout** again to kill those jobs.)

CTRL/q

CTRL/]

CTRL/x and then **CTRL/z**

Esc

:q!

CTRL/d

If all these commands fail, try to get a consultant or your support provider to help you logout. If no consultant is available and you are connected via a Macintosh or PC, you may reboot your workstation. If you are connected via a Unix workstation, do not reboot.

Commands to avoid

Certain Unix commands can be confusing (and sometimes dangerous) for beginners. If someone shows you a new command, be sure to ask if it might be dangerous to use. In particular, you should not experiment on your own with the commands **chmod**, **emacs**, or **vi**.

VIII. Technical details about IU's advanced IT facilities

The Research SP System

The Indiana University Research SP provides a theoretical peak compute capacity of approximately 1 trillion arithmetic operations per second (1 TeraFLOPS), with memory and storage scaled to match. As of this writing, the SP is one of the largest university-owned supercomputers in the US, and is ranked in 78th place on the list of the world's most powerful supercomputers (http://www.top500.org/). This system provides IU researchers with capabilities for advanced computations that few, if any, of your scientific peers at other institutions have.

The SP consists of large number of subcomponents called "nodes;" each node contains four to sixteen processors. There are a total of 144 nodes and 632 processors in the SP. The IBM SP runs IBM's version of Unix, called AIX. AIX is a very standard flavor of Unix and is well regarded in the computing industry.

The SP is subdivided into two units - the Aries complex and the Orion complex. The Aries complex is physically located in Bloomington, while the Orion complex will be moved to Indianapolis as soon as the installation of air conditioning facilities in the IUPUI machine room permit.

- The Aries complex is ideal for running serial SPSS, SAS, MATLAB, Mathematica, and Maple jobs, in addition to serial or parallel Fortran/C/C++ programs.
- The Orion complex contains nodes with up to 16 processors and up to 32 gigabytes of system memory, including one node containing IBM's newest Power4 processors. These processors are designed to provide excellent performance for both scientific and commercial/database workloads. The Orion complex provides an excellent symmetric multiprocessing (SMP) environment for large-memory parallel jobs.

The home page for the IU Research SP is:

http://sp-www.iu.edu/

A more detailed system description of the IU Research SP may be viewed at:

http://sp-www.iu.edu/TeraFLOP.SP.shtml

Find more information in the "How to use the IBM Research SP" section of this document.

The Sun E10000 System

The Indiana University Sun E10000's 64 processors and 64GB of memory constitute the Solar supercomputing system. It runs its own single-image copy of the Solaris (Unix)

operating system. Researchers will typically interact with the Sun E10000 in one of two contexts. If you use GCG, a set of programs and databases used to analyze molecular sequences, you will be using the E10000 transparently via a Web interface. There may also be researchers who have specific needs for working in a Solaris environment.

The home page for the IU Sun E10000 is:

http://www.indiana.edu/~rats/research/solar/solar.shtml

Some additional Sun E10K system information may be viewed at:

http://www.indiana.edu/~rats/research/solar/E10K.intro.shtml

Find more information in the "How to use the Sun E10000" section of this document.

The Massive Data Storage System

Many people are familiar with storing data by writing it to a Readable/Writeable CD. Such a CD holds about 600 MBs of data. Some advanced instruments are set to write the results of a particular analysis on a CD and then the CD is handed to the researcher. If you generate lots of data, there is some chance you have a pile of CDs lying around. This can be a headache and your data retrieval can be slow. First there is the time required to find the proper CD (which can be considerable), then the time to load the CD, and then the data is read from the CD on a typical microcomputer at a rate of 150-500 KBs/sec.

IU has a massive data storage system that consists of two large tape robots and a software system called HPSS (High Performance Storage System). One of the tape robots is located in Indianapolis, the other in Bloomington. The two tape robots have a combined storage capacity of 480 TBs (terabytes). To put this in context, one terabyte is roughly the capacity of 1,600 CDs. These tape robots hold thousands of individual data tapes, and each tape holds 20-60 GBs of data. HPSS is an example of a Hierarchical Storage Management (HSM) system since it uses a hierarchy of faster and expensive (disk) to slower and cheaper (tape) storage devices to give the appearance of massive storage capacities. When you send data to HPSS, it is first written to a fast disk cache (we currently have roughly a terabyte of disk). When not accessed for a time, your data migrates seamlessly to tape(s) in the tape robots. However, you can still do a directory listing since directory and file information live on disk forever. When you go to retrieve data, a nominal delay is incurred while a tape is located and mounted in the drive. Storage of data this way is called *nearline storage* to distinguish it from *online storage*, which allows instant access to data stored in a computer's random access memory chips or on spinning disks.

IU's unique geographically distributed storage system allows for tremendous data integrity, as it is possible to keep one copy of a data set in Indianapolis and another in Bloomington. HPSS was developed initially for US nuclear weapons laboratories, so it provides for data security and longevity in several other ways. Data can be encrypted as it is written so that a hacker cannot decipher the data even if he were to break in and physically steal a data tape. HPSS also protects against the fact that tapes do get old and

wear out. One type of protection is of course available through the keeping of two copies of data in Bloomington and in Indianapolis. HPSS will also write data from one individual tape to another either when the tape gets to a particular age, or reports (correctable) data read errors, so that even the individual copies of a data set in one location should be very secure. IU's massive data storage system was implemented with very long term data storage (several decades) in mind.

The IU Massive Data Storage System is easy to access, with a number of user interfaces that range from simple Web and FTP front ends to high performance parallel FTP from the IBM SP and the Sun E10000 supercomputers. It permits you to move data from the tape archive to the hard disk of your own workstation or to the disks of one of IU's supercomputers very quickly. It takes 30-90 sec (on average) to retrieve a data tape and load it into the robot's tape reader. Data is then read from the tape and delivered to your desktop over the local network at a rate of 0.3-20 MB/sec depending on how fast your network is.

Find more information in the section "How to use the Common File System or Massive Data Storage System."

IX. How to use the IBM Research SP

Logging in

One of the goals of the Research and Academic Computing Division of UITS is to provide robust, reliable services to the IU community. A critical part of making any computer system reliable is keeping it secure against hackers. In order to protect the security, and thus the reliability, of UITS research computing systems, all sessions must be encrypted using a protocol called Secure Shell (Version 2 or later).

If you do not have SSH installed on your personal workstation there are several options:

- Call your departmental computer support person for help getting SSH installed on your computer.
- If you use the Windows operating system, you may download SSH Secure Shell version 2.x for Windows from:

  http://www.itso.iu.edu/services/ssh/

  (You will need to provide your IU network ID and password). By default the downloaded file will be added to your desktop and called "SSHWinSecureShell24.exe". After it has downloaded, double-click on it and follow the directions that appear for installation. The installation program should put two icons on your desktop: one is labeled "SSH Secure Shell Client" and the other "SSH Secure File Transfer Client". These are pre-licensed for IU, so no further licensing steps are required on your part.

- If your personal workstation has the Unix operating system and does not have Secure Shell version 2.x installed, than please e-mail *ussg@iu.edu* for help in obtaining and installing Secure Shell version 2.x.
- If you use a version of Mac OS prior to OS X, you can download a freeware SSH2 client named MacSSH from:

  http://www.macssh.com/

  Follow the instructions at that site for easy installation. You will work with an SSH2 window as in Windows.
- If you run Mac OS X, SSH is installed by default and can be accessed through the terminal prompt.

To log into the SP from a Windows or Macintosh workstation, double-click the SSH client icon on your computer. When the program opens, you must specify a particular node to log in to. The SP nodes are named *aries01-127.ucs.indiana.edu* and *orion02-17.uits.iupui.edu* (node orion01 is reserved for UITS software development and testing). If you want to run a commercial package interactively you will have to log in to one of the nodes that has installed on it the software you want to use[1]. A list of the commercial software available on the SP, and the nodes on which this software is available, is available online at:

http://sp-www.iu.edu/SP.software.shtml

However, many people will generally be running software packages in batch mode. In this case it does not matter which node you log in to.

If your personal workstation has the Unix operating system or Mac OS X, to log into the SP you would generally enter at the prompt something like:

```
ssh nodename
```

If you wanted to log in specifically to the aries05 node, you would enter:

```
ssh aries05.ucs.indiana.edu
```

In any of the above cases, you will be asked for your password after you specify the node you want to log in to. Type the password, and then you'll see the Message of the Day and (except for the very first time you log in, discussed just below), you'll see the Unix `$` prompt.

[1] This situation results from the high cost of licensing software; commercial packages are licensed for enough nodes to meet the demand for that package, but no more, as licensing the commercial software on the SP for all nodes of the SP would cost millions of dollars per year.

Logging in to the SP for the first time

When you use a computer that runs the Unix operating system you interact with a piece of Unix known as the shell. The very first time you log in to the SP, you must log in to the node **aries01.ucs.indiana.edu** to specify which Unix shell you prefer. Unless you are an experienced Unix user and have a specific reason to do otherwise, we recommend that you select the Korn shell (referred to often as k-shell or ksh). Here are the steps you should go through the first time you log in to select the k-shell:

The first time you log into the SP, you will be presented with the following greeting:

```
Welcome to the UITS Research SP!

This program is run the very first time you log in
to aries01 to allow you to select your SP login shell.
If you are uncertain which shell to select, choose
ksh (Korn shell).

1) ksh
2) csh
3) bsh
4) bash
5) tcsh
6) quit
Select 1-6:
```

Type **1** and press enter. Your change will be confirmed:

```
Your login shell was changed to the Korn shell
Select 1-6:
```

Type **6** and press enter to exit the system. You will receive the message:

```
Connection to aries01.ucs.indiana.edu closed.
```

To verify your shell selection, log into *aries05.ucs.indiana.edu*. At the prompt type:

```
echo $SHELL
```

The response should be:

```
/bin/ksh
```

Logging out

To end your Unix session, enter one of the following:

`logout`

`exit`

CTRL\d

If the system responds with `There are suspended jobs`, simply type the command again. Be sure always to log out when you are done using the SP. Never leave an active session unattended for long periods of time.

Files and directories

Once logged in, you will be located in your home directory. A Unix directory is the same as a Windows or Macintosh folder.

A filename in Unix consists of a name and optional extensions. Filenames can be up to 255 characters long and have any number of extensions, but cannot contain certain special characters. Use only letters, numbers, dashes (-), underscores (_), and dots (.) in filenames. Unlike Windows and Macintosh, Unix does not associate files with particular programs. You may find it convenient to use extensions to keep track of your file types. For example, you could end all your Maple files with .maple, or all your Mathematica files with .math.

To **see a list of your files**, at the prompt enter:

`ls`

You will see a listing of your files and directories, similar to this:

```
Mail/   my-very-large-job.spss     myshellscript*
News/   myinput.sas                note-to-myself.txt
bin/    myjob1.sas
```

To **view the contents of a file** named *note-to-myself.txt*, type:

`more` ***`note-to-myself.txt`***

The contents of whatever file you named (in this example, it's *note-to-myself.txt*) will be displayed on your screen, one screen at a time. To go to the next screen, press the space bar. To go back one page, press **b**. To quit viewing the file before you get to the end, press **q**.

To **rename a file**, use the **mv** (move) command. For example, if you mistyped a filename and called it *myjo1.sas* when you meant *myjob1.sas*, you can correct this error by entering at the prompt:

```
mv myjo1.sas myjob1.sas
```

The file is renamed *myjob1.sas*, and there is no longer a file named *myjo1.sas*.

To **make a copy of a file**, use the **cp** command. For example, if you have a file *myjob1.sas*, and you want to make a copy called *myjob2.sas*, enter at the prompt:

```
cp myjob1.sas myjob2.sas
```

You now have two files with identical contents but different filenames, *myjob1.sas* and *myjob2.sas*.

To **delete a file**, use the **rm** command. For example, to remove the file *myshellscript*, you would enter at the prompt:

```
rm myshellscript
```

The system will ask: `remove myshellscript?` If you're sure you want to remove it, press **y**, or if you change your mind and want to keep it, press **n**.

To **create a new directory (folder)**, use the **mkdir** command. For example, to create a directory named *FitchCode*, you would enter at the prompt:

```
mkdir FitchCode
```

To **enter a subdirectory**, use the **cd** command. For example, to open the newly created *FitchCode* directory, enter at the prompt:

```
cd FitchCode
```

To return to the parent directory of the *FitchCode* directory, enter:

```
cd ..
```

Or, to directly return to your home directory, enter:

```
cd
```

Editing files on the SP

To edit files on the SP we recommend that you use a very simple Unix editor called Pico. Use Pico to create a new file or change the contents of an existing file. To edit a file named *note-to-myself.txt*, at the prompt, enter:

```
pico note-to-myself.txt
```

Pico either opens the file *note-to-myself.txt* (if it exists), or creates a new, empty file with that name. Pico fills your screen with the contents of the file *note-to-myself.txt* and

displays a command summary at the bottom of the screen. For example, the first entry in the command summary reads:

```
^G Get Help
```

This means that to get help, hold down the CTRL key while you press **g**.

Here are the basic Pico commands:

CTRL/v	move forward one screen
CTRL/y	move back one screen
CTRL/o	save (output) your file
CTRL/x	exit Pico (you'll get a chance to save your file)
CTRL/r	at the cursor insert another file's contents
CTRL/k	cut the cursor's line of text
CTRL/u	paste the previous cut
CTRL/w	search the file for a text string

To add text to your file, simply begin typing. Use the arrow keys to move the cursor.

Transferring files

Windows

If your personal workstation has the Windows operating system, then use the SSH Secure File Transfer Client to transfer files. To use FTP to transfer files in SSH Secure Shell for Windows, you'll first need to open a file transfer window. Double-click the SSH client icon on your computer, then, from the Window menu, select **New File Transfer**.

Once you have the file transfer window open, you'll need to connect to the computer you'd like to use to transfer files. Follow these steps:

1. From the **File** menu, select **Connect...**
2. In the **Host Name:** field, type the name of the host to which you are connecting (e.g., *aries05.ucs.indiana.edu*). In the **User Name:** field, type your username on that host. Click **Connect**. If this is the first time you have connected to this host, you will be asked if you want to save the new host key to the local database. Click **Yes**.
3. When you are prompted for a password, enter your password.

When the software connects to your host, you should see your directories on that server on the left side, and the files in the active directory on the right.

To move a file from your computer to the server, follow these steps:

1. From the **Operation** menu, select **Upload...**
2. An **Upload - Select Files** window will open. Browse to the file you'd like to move.
3. Click the **Upload** button.

To move a file from the server to your computer, follow these directions:

1. Find the file or folder on the server you'd like to download, and click it to highlight it.
2. From the **Operation** menu, select **Download...**
3. A **Download - Select Folder** window will open. Browse to the folder where you'd like to place a copy of this file.
4. Click the **Download** button.

Mac OS

If your personal workstation is a Macintosh, use MacSFTP to transfer files. Follow these steps:

1. Double-click the MacSFTP icon. The MacSFTP Login window will display fields for connecting to a remote computer.
2. In the **Host Name:** field, type the address of the remote host to which you wish to connect (e.g., *aries05.ucs.indiana.edu*).
3. In the **Login:** and **Password:** fields, type your username and password for the remote host.
4. If you want to log into a directory other than your home directory, type it in the **Path:** field (e.g., *www*).
5. Click the **Connect** button to start the SFTP connection.

   Note: The first time you connect to a host, SFTP may warn you that it can't determine the host's authenticity. Click **Yes** to accept the host's keys and continue connecting.
6. A window will open displaying the list of files on the remote host. To upload files or folders, drag them from Finder windows into the MacSFTP window. To download files or folders, drag them from MacSFTP into the Finder.

If you use Mac OS X, you can run SFTP from the terminal prompt with the command:

```
sftp nodename
```

Unix

If your personal workstation has a Unix operating system, use secure copy (**scp**) to transfer files:

To copy a file named *file.dat* from your home directory on the Research SP's aries05 node to the current directory of your Unix workstation, enter at the prompt:

```
scp username@aries05.ucs.indiana.edu:file.dat ./
```

To copy *file.dat* to your home directory on the Research SP's aries05 node from the current directory of your workstation enter at the prompt:

```
scp file.dat username@aries05.ucs.indiana.edu:
```

You will be prompted for your password. Then you will see an indication that the transfer was successfully completed.

Printing files

Once the file has been downloaded, follow these steps:

Unix

If you run Unix on your workstation, simply print the downloaded file as you usually would using **lp** or **lpr**.

Windows or Mac OS

If you run Windows or Mac OS on your workstation, the easiest approach is to open the downloaded file with WordPad (Windows), SimpleText (Mac OS) or Microsoft Word and print the file from within the application as usual. It is quite common to end up with a messy file because the text is wider than the present margins in your document. The easiest way to deal with this is to switch your printing to landscape mode, so that the text is printed lengthwise across the paper.

To print a document in landscape mode using Word 97 or 2000 for Windows, follow the steps below:

1. From the **File** menu, select **Page Setup**.
2. In the Page Setup window, click the **Paper Size** tab.
3. Under **Orientation**, select **Landscape**.
4. Click **OK**.

To print a document in landscape mode using Word 98 or 2001 for Mac OS, follow the steps below:

1. From the **File** menu, select **Page Setup...** .
2. Under **Orientation**, select the **Landscape** icon.
3. Click **OK**.

To ensure that the document is really going to print out properly, it is useful to look at the document with Microsoft's "Print Preview" feature.

Running Commercial Programs/Applications on the Research SP

SAS jobs on the Research SP

Sample SAS codes

A SAS program normally contains a data step and a procedure step.

For example, the code given below reads a data file, sample.dat, with six variables: id (identification number of subject), gender, wt_grp (weight classification, 1=underweight, 2=normal weight, 3= overweight), glucose (glucose level), bp (blood pressure classification, 1=normal, 2=high), and reactime (reaction time for visual stimulus), from the root directory.

```
DATA sample;
INFILE '~/sample.dat';
INPUT id 1 gender $ 3 wt_grp 5 glucose 7-9 bp 11
      reactime 13-15;

PROC PRINT;
RUN;

PROC ANOVA data=sample;
  CLASS gender;
  MODEL reactime=gender;
RUN;

PROC GLM data=sample;
 CLASS gender;
 MODEL glucose=gender;
RUN;

PROC REG data=sample;
  MODEL glucose=reactime;
RUN;

ENDSAS;
```

In the above example, the data file, *sample.dat*, is stored in the root directory of the user. If the data file is stored in another directory, then specify the full pathname, on the INFILE statement in the program file. For example,

```
INFILE '~/ingen/sample.dat';
```

tells SAS to find the data file and *sample.dat*, in the ingen subdirectory. The data file is stored in the directory in the following format:

```
1 m 1  99 1 210
2 f 2 320 2 420
3 f 2 195 2 350
4 m 1 110 1 215
5 m 2 218 2 364
6 f 3 120 1 355
7 m 3 125 1 335
```

Running SAS Jobs on the Research SP

Once you have both the program file and data file, to execute the job, at the SP System prompt, you may choose one of the following methods:

Non-Interactive process

To run your program interactively, at the system prompt, enter:

```
sas sample.sas
```

Once the job is executed without errors, two additional files will be written to your directory, *sample.log,* and *sample.lst*. The *.log* file contains log information regarding your job, and the *.lst* file will contains the output listing. If the *.lst* file is not created, examine the *.log* file for any error messages. If there are error messages, edit the program file with corrections, and rerun the job.

However, if you run a job interactively, your computer won't allow you to continue your work until the job is completed. The method of execution is only recommended if your job requires a short time to complete.

Background process

If you want to continue with your computing activities, while the job is running, then you can execute the job as a background process. To execute the job as a background process, type:

```
sas sample.sas &
```

Once the job is completed successfully, the .log, and .lst file will be written to the directory.

Batch process

If your SAS program will run for more than 20 minutes of processor time or if you need large amounts of memory, you must use the batch queue to run your program. To use the batch queue, run SAS using the command **sasjob** rather than with the **sas** command. For example, if you normally run **sas** ***sample.sas***, run **sasjob** ***sample.sas***. (Run **sasjob** from the directory that you want to be the current directory when the job actually runs.)

The batch queue will produce two files in addition to the usual log (*.log*) and list (*.lst*) files in the default directory. These files end with the suffices *.err* and *.out* (e.g., sample.err and sample.out). They will be typically empty unless you've used some unusual options to SAS that make use of the Unix concepts of standard output and standard input. 99.99% of users can ignore the *.err* and *.out* files.

By default use of the batch queue allows your job to consume 8 days of processor time. You can also request up to 1 gigabyte of memory. To request more than the default amount of memory (up to 1 gigabyte), use the **memsize** option. For example,

```
sasjob –memsize 1G sample.sas
```

requests 1GB of memory. You can change the memsize parameter as necessary, depending on the capacity of the machine (e.g., 500M, 750M, 1G). Larger amounts of memory are available, but you'll need to prepare specialized batch submission scripts to request them. Find details at:

http://sp-www.iu.edu/LLguide.shtml

To view sample script files for a number of statistical and mathematical software applications, after logging on to the SP, enter:

```
cd ~statmath/scripts
```

You may copy any file from the directory by using the command:

```
cp ~statmath/scripts/filename
```

(replace ***filename*** with the name of the file you want to copy).

A sample SAS program file and data file can be copied from the Research SP. Once you are logged on to the SP system, type:

```
cp ~statmath/scripts/clas.sas .
cp ~statmath/scripts/clas.dat .
```

The line-by-line explanation of the code used in this program can be found at:

http://www.indiana.edu/~statmath/stat/sas/unix/3.html

Additional sample files provided by the SAS vendor can be viewed/copied on the SP node aries05 in the directory:

/statapps/sas8.2/samples

For more information on using SAS, or any other statistical and mathematical computing software, contact the UITS Stat/Math Center at *statmath@indiana.edu*, 812-855-4724, 317-278-4740, or visit:

http://www.indiana.edu/~statmath/

SPSS jobs on the Research SP

Sample SPSS codes

The SPSS code given below, reads a data file, sample.dat, with six variables: id (identification number of subject), gender, wt_grp (weight classification, 1=underweight, 2=normal weight, 3= overweight), glucose (glucose level), bp (blood pressure classification, 1=normal, 2=high), and reactime (reaction time for visual stimulus), from the root directory. Suppose this code is stored in a file, sample.sps, in the root directory.

```
DATA LIST  FILE=sample.dat
  /id 1 gender 3 (A) wt_grp 5 glucose 7-9 bp 11 reactime 13-15

LIST VARIABLES gender wt_grp glucose

ONEWAY  glucose BY wt_grp (1,3)

REGRESSION
  /DEPENDENT=reactime
  /METHOD=ENTER glucose

FINISH
```

If the data file is stored in another directory, then specify the full pathname, on the `FILE` statement in the program file. For example,

```
DATA LIST FILE '~/ingen/sample.dat';
```

tells SPSS to find the data file, *sample.dat*, in the ingen subdirectory. The data file is stored in the directory in the following format:

```
1 m 1  99 1 210
2 f 2 320 2 420
3 f 2 195 2 350
4 m 1 110 1 215
5 m 2 218 2 364
6 f 3 120 1 355
7 m 3 125 1 335
```

Running SPSS Jobs on the Research SP

Note: You must be logged into the node aries05 to run SPSS.

Once you have both the program file and data file, to execute the job, at the SP System prompt, you may choose one of the following methods:

Non-Interactive process

To run your program interactively, at the system prompt, type:

```
spss -m sample.sps > sample.out
```

Once the job is executed, the output file, *sample.out*, will be stored in the default directory. Examine the file for any error messages also, as SPSS doesn't generate a separate log file. If there are error messages, edit the program file with corrections, and rerun the job.

However, if you run a job interactively, your computer won't allow you to continue your work until the job is completed. The method of execution is only recommended if your job requires a short time to complete.

Background process

If you want to continue with your computing activities, while the job is running, then you can execute the job as a background process. To execute the job as a background process, type:

```
spss -m sample.sps > sample.out &
```

Once the job is completed successfully, the *.out* file will be written to the directory.

Batch process

If your SPSS program will use more than 20 minutes of processor time, you need to submit your program to the batch queue. To do so, just run your program with the command **spssjob** rather than **spss**. For example, you might use the command**:**

```
spssjob -m sample.sps -output sample.out
```

The output will be stored in a file named *sample.out*.

To view the directory where a number of sample script files are stored for various UITS supported statistical and mathematical software, type:

```
cd ~statmath/scripts
```

You may copy any file from the directory by using the command:

```
cp ~statmath/scripts/filename
```

(replace ***filename*** with the name of the file you want to copy).

A sample SPSS program file and data file can be copied from the Research SP. Once you are logged on to the SP system, type:

```
cp ~statmath/scripts/clas.sps .
cp ~statmath/scripts/clas.dat .
```

The line-by-line explanation of the code used in this program can be found at:

http://www.indiana.edu/~statmath/stat/spss/unix/2.html

Additional sample files provided by the SPSS vendor can be viewed/copied on the SP node aries05 in the directory:

/statapps/spss6.1.4/data

For more information on using SPSS, or any other statistical and mathematical computing software, contact the UITS Stat/Math Center at *statmath@indiana.edu*, 812-855-4724, 317-278-4740, or visit:

http://www.indiana.edu/~statmath/

Matlab jobs on the Research SP

There are three ways to run a Matlab program on the SP System: interactively, as a background process, or by submitting the program to the LoadLeveler as a batch process.

Interactive process

This is just typing the commands one-by-one at the Matlab command line. At the Unix prompt you can start Matlab with the command:

```
matlab
```

Your terminal will display the Matlab prompt:

```
                    < M A T L A B >
        Copyright 1984-2001 The MathWorks,Inc.
           Version 6.1.0.450 Release 12.1
                      May 18 2001

    >>
```

You can now enter Matlab commands. For example, to produce a four-by-four Hilber matrix you can enter at the prompt:

```
hilb(4)
```

with the following output:

```
ans =

    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

Background process

As with all Unix systems you can start a program and run it in the background. To do this you first need to make a text file that has the commands you want Matlab to run. In this example we will load a signal, assumed to have been sampled at 100Hz, from the file *signal.mat*. We then apply a third-order 30Hz lowpass Butterworth filter and save the result in a file called *result.mat* . Using a text editor such as pico you can make the following program and save it as *matlabinput*.

```
load signal x

[b,a]=butter(3,30/50);

y=filter(b,a,x);
save result y
quit
```

The following command will feed these commands to Matlab and display Matlab's output on the terminal:

```
matlab < matlabinput
```

However, it is more common to have the output written to some other file. The following command will write Matlab's output to a file called *matlaboutput* and write any operation system errors to a file called *matlaberror*.

```
matlab < matlabinput > matlaboutput 2> matlaberror &
```

The ampersand at the end of the line tells the computer to run matlab in the background so you can still use your terminal. Please, note that only operating system errors are written to *matlaberror*. For example, if a file is missing, this will show up in *matlaberror*. Errors that are handled within Matlab, such as dividing by zero, aren't noted here but in *matlaboutput*. Memory errors are a special case and depending on the exact nature of the error they might show up in either file.

After running the program you can look in your directory. There should now be text files called *matlaboutput,* and *matlaberror*. The file *matlaberror* should be empty. There will also be a file called *results.mat* that records the value of *y* in a format Matlab can load later.

Batch process

Any Matlab program that takes more than twenty minutes of time should be submitted to the batch queue. To submit a batch job use the command **matlabjob** rather than the command **matlab**. You will also have to specify the name of the input file using the command line option **-input**. For example, use:

```
matlabjob -input matlabinput
```

The output and error will be stored in files named from the input file containing suffices .out and .err, respectively. For, example if your input file is *matlabinput*, output will be in *matlabinput.out* and error will be in *matlabinput.err*. Optionally, you can specify names for the output and error files with the **-output** and **-error** options, respectively. For example:

```
matlabjob –input matlabinput –output matlaboutput -error matlaberror
```

For more details about checking on the status of your job, see the section "More about LoadLeveler" below.

This job won't take too long to finish. When it's done you should see files called *matlaboutput* and *result.mat* in the starting directory.

For a brief introduction to the syntax and capabilities of Matlab please see the Stat/Math Center's introductory guide at:

http://www.indiana.edu/~statmath/math/matlab/gettingstarted/

For more detailed questions about Matlab or its applications you can contact the UITS Stat/Math Center at *statmath@indiana.edu*, 812-855-4724, 317-278-4740, or visit:

http://www.indiana.edu/~statmath/

Running your own FORTRAN programs, C programs, or other large jobs using LoadLeveler

For large jobs (those requiring more than 20 minutes of CPU time) you are required to use a batch job management facility called LoadLeveler. For the sake of ease of reading, this document tells you how to run a FORTRAN, C or C++ program already written for the SP first, and then tells you how to compile a FORTRAN, C, or C++ program on the SP. Information is presented in this order in the belief that you will likely run someone else's homebrew FORTRAN or C program before you write your own.

Running a precompiled FORTRAN, C, or C++ program on the Research SP

Script files to run SAS, SPSS, and Matlab jobs using LoadLeveler are stored on the SP system. To copy the files to your default directory, at the SP system prompt, type:

```
cp ~statmath/scripts/filename
```

(replace ***filename*** with **sas.script**, **spss.script**, or **matlab.script**). To copy program files stored in the same directory, refer to the section on SAS, SPSS, or Matlab in this document.

Compiling FORTRAN, C, C++ programs on the Research SP

Before you run a FORTRAN, C, or C++ program on the SP, the source code must be compiled using the SP's compiler. Except for very large programs, this is easily done interactively. To illustrate compiling jobs and submitting them to LoadLeveler, you may look at some elementary examples in the hpc directory e*xamples*. To view the contents of this directory, at your SP prompt type:

```
ls ~hpc/examples
```

(Don't forget the leading "~"!) You will see:

```
helloWorld.C          helloWorld_C++*       helloWorld_F77.out
helloWorld.c          helloWorld_C++.out    llScript_C
helloWorld.f          helloWorld_C.out      llScript_C++
helloWorld_C*         helloWorld_F77*       llScript_F77
```

The Fortran 77 example program is *helloWorld.f*, the C example is *helloWorld.c*, and the C++ example is *helloWorld.C*. To view the Fortran code, at the prompt, enter:

```
more ~hpc/examples/helloWorld.f
```

Do likewise for the C and C++ programs. To experiment with these programs, at the prompt type **cd** to enter your home directory, then type:

```
mkdir examples
```

to create your own directory by that name. Then type **cd examples** to enter your new directory. You may copy the example Fortran program to your *examples* directory by typing at the prompt:

```
cp ~hpc/examples/helloWorld.f ./
```

Type **ls** to see the file *helloWorld.f* listed.

The executables in the *~/hpc/examples* directory are identified by the trailing "*". You may now compile your own executables. After copying the Fortran program, at the prompt enter:

```
xlf –o helloWorld_F77 helloWorld.f
```

To compile the C program, enter:

```
xlc –o helloWorld_C helloWorld.c
```

To compile the C++ program, enter:

```
xlc –o helloWorld_C++ helloWorld.C
```

You may run each of the executables interactively by typing each of the following at the prompt:

```
./helloWorld_F77
./helloWorld_C
./helloWorld_C++
```

If you have a program that runs a long time (more than 20 minutes), you will want to submit the executable to LoadLeveler for batch processing. A LoadLeveler script makes this easy. The example LoadLeveler scripts in the *~/hpc/examples* directory are named *llScript_F77*, *llScript_C*, and *llScript_C++*. To copy the LoadLeveler script for the *helloWorld_F77* executable, type at the prompt:

```
cp ~hpc/examples/llScript_F77 ./
```

Use **more** to view this script or **pico** to edit it. To submit the *helloWorld_F77* executable to LoadLeveler, at the prompt type:

```
llsubmit llScript_F77
```

You may do likewise for the C and C++ scripts.

To check the status of your submission, at the prompt type:

```
llq -u username
```

where ***username*** is your username. A table will appear that looks like this:

```
Id                     Owner      Submitted   ST PRI Class    Running On
aries02.5285.0         username   5/23 09:28   I  50  a
```

The status `ST` shows an `I`. Your job is idle, waiting to run. Reenter "**llq -u *username***" (perhaps several times) and you will soon see the `ST` column showing an `R`. Your job is running. Finally another reentry of **llq -u *username*** will show a `C` in the `ST` column, indicating your job has been completed. Now enter **ls** and you will see a new file named *helloWorld_F77.out*. This is your program's output file. Use **more** to view its contents, which you will see matches the standard output you obtained earlier from the interactive run.

This process is detailed more thoroughly in the section "More about LoadLeveler" below.

If your source code consists of more than one file, or requires compiler options such as optimization to improve run time, then please contact the HPC high performance computational science staff at *hpc@indiana.edu* to obtain assistance.

If your research can benefit from improving the performance of your program by parallelizing your source code, then again please contact the HPC high performance computational science staff at *hpc@indiana.edu*. There is no cost or charge to you for any computational developmental work undertaken for you by HPC.

More about LoadLeveler

LoadLeveler is an IBM software product which is used on SPs and other clustered systems to manage batch jobs. Users interact with LoadLeveler via a graphical user interface or via commands. LoadLeveler has a default job scheduler but also interfaces with external schedulers. On the Research SP, we use the Maui Scheduler for its advanced functionality that allows fairshare, advance job reservations, resource banking, quality of service, and more. Commonly used LoadLeveler and Maui commands and their functions are:

llsubmit	Submits your back script to the batch queues
llq	Queries the status of running and queued jobs
llconfig	Displays the current configuration of LoadLeveler machines
llstatus	Displays the status of LoadLeveler machines
llcancel	Cancels a running or queued job
llclass	Describes the defined batch classes (queues)
showq	Shows information about job scheduling
showfairshare	Displays fairshare targets and balances
showres	Displays job reservations
xloadl	Invokes the graphical user interface

To give you more familiarity with LoadLeveler, let's take a look at a typical user interaction in which a submitted job is queried and allowed to run to completion or is cancelled by the user. Let's assume user *jdoe* was logged into node aries09 and submitted a job to run a serial Fortran program. Typing **llq** would display all the currently running and queued jobs for all users, but to just look at his own jobs, *jdoe* would type

llq –u *jdoe*

and get the following output:

```
ID                  Owner       Submitted     ST   PRI  Class      Running On
------------------------------------------------------------------------------
aries09.3034.0      jdoe        4/22 14:46    R    50      a       aries02
```

This shows that the job was submitted from aries04 at 2:46pm on 4/22 by user *jdoe* to batch class *a*. The `ID` is the jobid assigned by LoadLeveler and can be used as an argument to LoadLeveler commands. The `ST` shows the current job status, which can be one of the following:

R	running	RM	removing
ST	starting	C	completed
I	idle	CA	cancelled

Although the job was submitted from aries09, it is actually running on aries02, which runs class *a* jobs. Now let's assume the user realized after the job was submitted that it might need more time to run than the limits defined for class a jobs. He could check time remaining before his job reached the class a time limit by typing:

```
showq
```

This would yield the following output:

```
JOBNAME          USERNAME   STATE          PROC   REMAINING   STARTTIME

aries09.30304.0 jdoe        Running     1   1:21:29:59   Mon Apr 22 14:57:06
```

Let's assume now that the user realizes that one day, 21 hours, 29 minutes, and 59 seconds will not be adequate time for the job to complete, so he decides to cancel this running job and submit it to the class *b* job queue instead. He would simply cancel the job with the llcancel command:

```
llcancel aries09.30304.0
```

Or, if he had several jobs running and wanted to cancel them all, he could type:

```
llcancel -u jdoe
```

Then he would revise his job script to say **class=*b*** instead of **class=*a*** and resubmit the job.

When a job completes normally, is cancelled, or is aborted due to errors, LoadLeveler sends an e-mail message to the user at the host where the job was submitted. This message displays the final job status and the job return code, the name of the node on which the job ran, and the user and system time consumed by the job. (If you want to have this LoadLeveler e-mail forwarded to the host where you normally receive mail, be sure to set up a .forward file, which must include your full e-mail address, in your SP home directory.) On the following page is a typical LoadLeveler e-mail resulting from a job which ran to completion.

```
Date: Mon, 22 Apr 2002 17:57:27 -0500
From: LoadLeveler <loadl@aries09.ucs.indiana.edu>
Message-Id: <200204122256.RAA65050@aries09.ucs.indiana.edu>
To: jdoe@aries09.ucs.indiana.edu
Subject: aries09.ucs.indiana.edu.3225

From: LoadLeveler

LoadLeveler Job Step: aries09.ucs.indiana.edu.3225.0
       Executable: /N/u/jdoe/SP/bin/myprog
       Executable arguments:
       State for machine: aries02.ucs.indiana.edu
       LoadL_starter: The program, myprog, exited normally and returned
       an exit code of 99.

This job step was dispatched to run 1 time(s).
This job step was rejected by Starter 0 time(s).
Submitted at: Mon Apr 22 15:46:06 2002
Started at: Mon Apr 22 15:47:06 2002
Exited at: Mon Apr 22 17:57:08 2002
Real Time: 0 02:10:02
Job Step User Time: 0 02:00:01
Job Step System Time: 0 00:10:01
Total Job Step Time: 0 02:10:02
```

For more information about LoadLeveler and Maui commands, consult the man pages or see the online documentation at:

http://sp-www.iu.edu/Llguide.shtml

IBM Bioinformatics Tools on the SP

A mirror of IBM's Bioinformatics and Pattern Discovery Web server is available on the IBM SP. The server provides tools and data that are united through the use of the TEIRESIAS algorithm for pattern discovery. The server is available at:

http://orion17.uits.iupui.edu/.

To most effectively use the tools, read the references that are provided at the bottoms of the Web pages. The on-line tutorial is not recommended.

TEIRESIAS is an algorithm that locates recurrent patterns within sequences. The sequences can be molecular sequences, words, or anything that can be encoded as a stream of integers. Patterns are variable in length and content. You specify the maximum length of pattern and the minimum number of positions in the pattern that must have values that are consistent among occurrences of the pattern. TEIRESIAS reports all such patterns that occur a minimum number of times that you specify. Reported patterns are maximally specific. That is, patterns with variable positions are not general specifications of other patterns of the same length that have more positions with fixed values.

MUSCA is an algorithm that aligns multiple sequences. It works in two steps. In the first, common motifs, areas of local similarity, are sought across sequences (using TEIRESIAS), and in the second the set of motifs is culled to produce an alignment. One advantage of the algorithm is that the alignment of a set of sequences is independent of the order in which sequences are provided to the algorithm.

Bio-Dictionaries are annotated genomes of many organisms. Briefly, they were built by using the TEIRESIAS algorithm to locate recurrent patterns in amino acid sequences from public databases. The recurrent patterns were annotated using annotations from public databases that describe the locations where the patterns occur.

The methods that were used to build the Bio-Dictionaries are also available for you to annotate protein sequences using the "Protein Annotation" engine. A separate engine is available for G-Protein Coupled Receptors.

TEIRESIAS can be used to discover associations among variables in data matrices in which columns represent variables or measured features and rows represent observations or subjects in which features were measured. For categorical variables (e.g., tree species, leaf type), associations are detected between particular values of one variable and particular values of others. Continuous variables can be analysed if they are represented using discrete categories. For example, a negative correlation between two variables that are represented by the categories high, medium, and low would be manifested as three associations: high of variable a with low of variable b, medium with medium, and low of variable a with high of variable b. The method has been dubbed "Association Discovery" and can be applied to the analysis of gene expression microarrays.

Finally, it is possible to perform Comparative Molecular Moment Analysis (CoMMA) and to search for patterns in the results using the TEIRESIAS algorithm.

X. How to use the Sun E10000

Using GCG and SeqWeb on the Sun E10000

GCG is a set of programs installed on the Sun E10000 supercomputer for the analysis of molecular sequences. GCG programs are available for database search and retrieval, sequence comparison, DNA/RNA secondary structure, evolution, fragment assembly, gene finding and pattern recognition, mapping, primer selection, protein analysis and translation. Compute-intensive programs take advantage of multiple processors on the supercomputer. These programs include FrameSearch, BLAST, FastA, FastX, TFastA, TFastX, and Ssearch. A complete list of programs is available at:

http://www.accelrys.com/products/gcg_wisconsin_package/

GCG sequence databases are updated bimonthly.

A Web-based portal to GCG known as SeqWeb is available. SeqWeb supports many GCG programs although not some of the least frequently used programs. To use it you need a SeqWeb account on the Sun E10000. Refer to the "How to request accounts" section of this guide for details. Once you have an account, see:

http://solar.uits.indiana.edu:8000/

Documentation for GCG is on reserve in the medical school library in Indianapolis as well as in the life sciences library in Bloomington. GCG commands are also available from the command line for users with intensive needs. Details on initializing the GCG command-line on Solar are available at:

http://www.indiana.edu/~rac/bioinformatics/gcgcmdline.html

Using NCBI BLAST on the Sun E10000

Facilities are available for using NCBI BLAST to query hundreds, thousands or tens of thousands of sequences against databases, using multiple processors on the Sun E10000 supercomputer. To do so, load sequence files onto the machine and run a command which identifies the input file, the type of search to do, the database to be searched and a destination for the output. For example, the command

```
blastjob -a 4 -p blastp -i mydata -d nr -o results -cpuhours 120
```

requests 4 processors (`-a`) to query peptide sequences (`-p`) in a file named `mydata` (`-i`) against the non-redundant database of amino acid sequences (`-d`) and place the output in a file named `results` (`-o`). The program is expected to consume 120 processors hours, which with 4 processors is about 30 hours of wall-clock time.

BLAST produces huge quantities of output. For example, the default output from thousands of queries is measured in gigabytes (thousands of megabytes). Options are available to reduce output. BLAST itself can produce tab-delimited output, and it is possible to request specific fields.

The BLAST algorithm uses multiple processors relatively inefficiently. To efficiently use many processors, the set of query sequences can be subset, and separate BLAST process can be started for the subsets. Scripts are available for subsetting datasets and for starting many BLAST processes.

Details on high-volume BLAST searching are available at:

http://www.indiana.edu/~rac/bioinformatics/solarblast.html

XI. How to use the Common File System or Massive Data Storage System

Once you have determined the storage service that best fits your needs, you may request a CFS or an MDSS account. Use the following criteria as a guide:

- The **CFS** is the right service for you if you need to store modest amounts of data (in the form of either small or large files) which must be accessed frequently, and from a variety of environments at IU, for example from the central research systems (SP, Solar/Lunar, Steel), on-campus Unix labs at IUB (DaVinci, Nations, or Ships), or from your personal workstation (running Windows, Mac OS, or Unix).
- The **MDSS** is the right service for you if you require large-scale storage (tens or hundreds of GB or more) that is arranged in large files (over 50-100MB) that you need to access relatively infrequently after you store them. In other words, the MDSS is ideal for data archival purposes or for near line access to your data.

Requesting accounts at IUPUI, IUK, or IUS

Requesting a CFS account at IUPUI, IU Kokomo, or IU Southeast

1. Connect to the IUPUI Network ID Services Web page at:

   https://iupui-accts.iupui.edu/

2. Click **Student Network ID Services** or **Faculty/Staff Network ID Services**, depending on your status.
3. Click **Creating additional network services on IUPUI computers**.
4. Log in with your Network ID and password. . (Some IUK users may not be able to get in since their network ID password is not synchronized with their IUK Exchange e-mail password. To get these in sync, please contact your local helpdesk.)
5. You should see the **Common File System (CFS)** option among the accounts you can request. Click it.
6. Now click **Yes, you are sure you want a Common File System (CFS) account**.
7. On the next page, enter your IUPUI Network ID password. Now click the **Create CFS Account** button. This should create your CFS account within minutes.

Requesting an MDSS account at IUPUI, IU Kokomo, or IU Southeast

1. Connect to the IUPUI Network ID Services Web page at:

   https://iupui-accts.iupui.edu/

2. Click **Student Network ID Services** or **Faculty/Staff Network ID Services**, depending on your status.
3. Click **Creating additional Network Services on IUPUI Computers**.
8. Log in with your Network ID and password. . (Some IUK users may not be able to get in since their network ID password is not synchronized with their IUK Exchange e-mail password. To get these in sync, please contact your local helpdesk.)
4. If you are a faculty or staff member or a graduate student, you should see **Massive Data Storage Service (MDSS)** as one of the accounts you can request. Click it. (If you are an undergraduate, you must find a faculty sponsor for a specific research project and have the sponsor e-mail *store-admin@iu.edu* with reasons for the account request.)
5. Click the **Fill out Form** button.
6. Fill out the form seeking information on how you intend to use the account. The form will ask you for the following:
    - your full name
    - your Network ID username
    - the project name (if any)
    - an estimate of how much storage space you will need immediately
    - an estimate of how much storage space you will need within 6 months
    - an estimate of how much storage space you will need within the next 12 months and beyond
    - duration of the storage (1 year, 3 years, indefinite)
    - preferred mode of access (ftp, pftp, hsi, DFS). Refer to the section "Accessing the CFS or MDSS" below for more information about these options.
    - amount of bandwidth required (1MB/s, 10MB/s, 100MB/s)
    - any special requirements (e.g., a private class of service and a private pool of tapes)

    Be sure to provide your preferred e-mail address on this form since we will need to contact you at this address when your account is set up.
7. Click **submit**.
8. On the next page, type in your Network ID password, then click **Create Account**. Your Network ID password will also become your MDSS password.

You will be sent an e-mail informing you when your MDSS account is ready for use.

Requesting Group/Lab/Departmental CFS/MDSS accounts at IUPUI, IU Kokomo, or IU Southeast

In order to establish group, lab, or departmental accounts, you will need to request the appropriate group network ID. Please contact the UITS Support Center (by calling 317-274-HELP, e-mailing them at support@iupui.edu, or visiting them in ES building room 2126) for further information on how to do this. Once you have the group network ID and the password in hand (it is usually snail-mailed to you by UITS), you can follow the procedure listed earlier to request a CFS or MDSS account.

Requesting accounts at IUB, IUE, or IUN

Requesting a CFS account at IU Bloomington, IU East, or IU Northwest

1. Connect to the IUB Network ID Services Web page at:

   http://accounts.ucs.indiana.edu/

2. Log in with your Network ID and password.
3. Click **Create accounts on UITS computers**.
4. You should now see **Common File System (CFS)** as an option among the accounts you can request. Click it.
5. Now click the **Yes, you are sure you want Common File Service (CFS) account** button.
6. On the next page, enter your Network ID password at the top and leave the remaining two fields blank so that your CFS password is the same as your Network ID password.
7. Click the **Create CFS account** button. This should create your CFS account within minutes.

Requesting an MDSS account at IU Bloomington, IU East, or IU Northwest

1. Connect to IUB Network ID Services Web page at:

   http://accounts.ucs.indiana.edu/

2. Log in with your Network ID and password.
3. Click **Create accounts on UITS computers**.
4. Click **Research-only Systems**.
5. If you are a faculty, staff, or a graduate student, you should see **Massive Data Storage Service (MDSS)** as an option among the accounts you can request. Click on it. (If you are an undergraduate, you must find a faculty sponsor for a specific research project and have the sponsor e-mail *store-admin@iu.edu* with reasons for the account request.)

6. Click the **Fill out Form** button. This will take you to a page where you must fill out a form. The form will ask you for the following information:
    - your full name
    - your Network ID username
    - the project name (if any)
    - an estimate of how much storage space you will need immediately
    - an estimate of how much storage space you will need within 6 months
    - an estimate of how much storage space you will need within the next 12 months and beyond
    - duration of the storage (1 year, 3 years, indefinite)
    - preferred mode of access (ftp, pftp, hsi, DFS). Refer to question 3 below for more information about these options.
    - amount of bandwidth required (1MB/s, 10MB/s, 100MB/s)
    - any special requirements (e.g., a private class of service and a private pool of tapes)
7. Be sure to provide your preferred e-mail address on this form since we will need to contact you at this address when your account is set up.
8. When you click **submit**, you will be taken to a page where you must enter your Network ID password. This will also become your MDSS password.
9. Click the **Create Massive Data Storage Service account** button.

You will be sent an e-mail informing you when your MDSS account is ready for use.

Requesting Group/Lab/Departmental CFS/MDSS accounts at IU Bloomington, IU East, or IU Northwest

In order to establish group, lab, or departmental CFS or MDSS accounts, the account sponsor should visit the departmental account request page at:

https://www.indiana.edu/~ithelp/dept.cgi

so that the appropriate group/lab/dept. Network ID can be created. Once you have the account password in hand (it is usually snail-mailed to you by UITS), you can follow the procedure listed earlier to request a CFS or MDSS account using this Network ID.

Armed with the appropriate account, you will be able to access the CFS and/or the MDSS services from your desktop workstation (running Unix, Windows, or Mac OS) as well as from the Research SP or Sun E10000 supercomputers.

Accessing the CFS or the MDSS

A variety of methods are available to you for storing and accessing data to/from the CFS or the MDSS. Which method is appropriate depends on the data transfer performance you desire and on file size.

Accessing the CFS from your Desktop PC/Mac without installing a client

This is the easiest method to access the CFS and is recommended for general file access (for file sizes less than 100MB). However, it is not the fastest method available (see "Accessing the CFS or MDSS via a DFS client from your Unix/Windows NT4/2000 PC" section below for a faster method).

Mapping a drive in Windows

In Windows ME/NT/2000/XP, you can map a drive letter to the CFS as follows:

1. Click the **My Computer** icon on the desktop.
2. Choose **Map Network Drive**.
3. Enter **\\cfs.iu.edu*username*** (where *username* is your username) in the **Folder:** box.
4. Click **Finish**.

If you are not already logged into the ADS domain, a dialog box will prompt you for your username and password. Once this is done, the CFS will appear to you as a network drive (for example, drive G:). In Windows 98, you must be logged into the ADS domain or into your PC with the same username and password as your CFS username and password to be able to do this.

Connecting via AppleShare IP in Mac OS

In Mac OS, connect via AppleShare IP to the server **cfs.iu.edu** and then choose the share with the same name as your username among the three choices you are offered (the other two are "SHARE" and "SCRATCH"). Authenticate with your CFS username and password.

Accessing CFS via the Web

You can also access the CFS via the Web by connecting to:

https://cfs.iu.edu/username

Authenticate with your CFS username and password.

Accessing the MDSS from your Desktop PC/Mac without installing a client

Using FTP to access the MDSS

The best data transfer performance between a Windows PC or a Mac and the MDSS is obtained via plain FTP. You can use any FTP client , such as the graphical clients WS_FTP in Windows and Fetch or Transmit in Mac OS on your machine to access the MDSS by connecting to the server *hpss.iu.edu*. (Please note that it is not possible to connect to hpss.iu.edu with secure FTP. Use the method described later, under the "Accessing the CFS or the MDSS via Secure FTP" section for secure FTP access.)

In the simplest case under Windows, you can open a command window and type at the prompt:

`ftp hpss.iu.edu`

You will be prompted for your MDSS username and password. Once you enter this information, you will be at the `ftp>` prompt. To see your files stored on the MDSS, type **dir** at the `ftp>` prompt. To download a file from the MDSS to the current folder on your PC/Mac's hard disk, enter **get filename**. To upload a file from your PC/Mac's hard disk to the MDSS, enter **put filename** (to transfer a binary file, such as a .EXE file, you must type **bin** at the `ftp>` prompt first).

Mapping a drive in Windows

For files less than 100MB in size (for larger files, the performance will drop substantially), you can map the MDSS as a drive letter under Windows ME/NT4/2000/XP as follows:

1. Click the **My Computer** icon on the desktop.
2. Choose **Map Network Drive**.
3. Enter **\\mdss.iu.edu*username*** (where *username* is your username) in the **Folder:** box.
4. Click **Finish**.

If you are not already logged into the ADS domain, a dialog box will prompt you for your username and password. Once this is done, the MDSS will appear to you as a network drive (for example, drive G:). In Windows 98, you must be logged into the ADS domain or into your PC with the same username and password as your MDSS username and password to be able to do this.

Connecting via AppleShare IP in Mac OS (to transfer files smaller than 100MB)

In Mac OS, connect via AppleShare IP to the server **mdss.iu.edu** and then choose the share with the same name as your username. Authenticate with your MDSS username and password.

Accessing MDSS via the Web

For file sizes less than 100MB, you can also access the MDSS via the Web by connecting to:

https://mdss.iu.edu/username

Authenticate with your MDSS username and password.

For more information on the MDSS, search for keyword **mdss** at:

http://kb.iu.edu/

Note: The client-less methods described here are also ideal for just perusing through the directory/folder listing in your CFS or MDSS space.

Accessing the CFS or MDSS via a DFS client from your Unix/Windows NT4/2000 PC

A DFS client allows file transfer to occur faster than methods described earlier that do not require a client to be installed. It is also a recommended method for transferring large files to the CFS or the MDSS (however, please note that FTP gives the fastest performance for accessing the MDSS).

Accessing the CFS or MDSS using a client-full, file system interface makes files placed in the CFS and MDSS visible to your workstation as an extension of its native file system, for example as drive G: under Windows NT4 or Windows 2000. The difference from the client-less method to map a drive discussed earlier is that a full client allows for better performance and makes a full suite of native file system utilities available to you.

To use this method, you can request a Distributed File System (DFS) client installation by sending e-mail to store-admin@iu.edu. Clients are available at no charge for use on IU machines by faculty, staff, and graduate students for the following operating systems: AIX 4.x, HP-UX 10.20, IRIX 6.5.x, Solaris 2.6-2.8, Tru64 UNIX 4.x, Windows NT, and Windows 2000.

Accessing the CFS or the MDSS via Secure FTP

If you have a secure FTP client such as the SSH Secure File Transfer client for Windows or MacSFTP for Mac OS, you can connect to the CFS or the MDSS by connecting to the

server **cfs.iu.edu** or **mdss.iu.edu** appropriately. Please note that data transfers over a secure FTP connection will be significantly slower than for plain FTP due to the stream encryption overhead in secure FTP. Learn more about secure FTP by searching for the keywords "secure ftp" in the IU Knowledge Base at:

http://kb.iu.edu/

Accessing the CFS or the MDSS from the IBM SP or Sun E10000

Once you log in to the SP or the E10000, type at the Unix prompt:

dce_login *username*

where *username* is your username. You will then be prompted to enter your CFS/MDSS password. Once you supply the password, you will have obtained the necessary credentials that allow you access to your CFS and/or your MDSS areas. To do so, simply use the **cd** (change directory) command at the Unix prompt. To access your CFS space, you would enter at the Unix prompt:

cd /:/home/*firstletter/secondletter/username*

To access your MDSS space, you would enter at the Unix prompt:

cd /:/mirror/*firstletter/secondletter/username*

where *firstletter* and *secondletter* are the first and second letters of your username. As an example, if your username is *joeuse*r, you would enter your CFS area by typing:

cd /:/home/*j/o/joeuser*

You would enter your MDSS area by typing:

cd /:/mirror/*j/o/joeuser*

Higher performing interfaces to the MDSS are also available.

Obtaining Help on the CFS or the MDSS

General help on using the CFS and the MDSS is available by searching for the keywords **cfs** and **mdss** at the IU Knowledge Base at:

http://kb.iu.edu/

or in the Distributed Storage Services Group's pages at:

http://storage.iu.edu/services.html

For further help with CFS or MDSS, please e-mail *store-admin@iu.edu.* Also, user training on massive data storage is available periodically from the Distributed Storage Services Group. For more information, please visit:

http://storage.iu.edu/edu.html

XII. How to use advanced visualization facilities

The following information is current as of May 2002. Please check our Web site for the latest information and updates on hardware and software technologies and projects:

http://www.avl.iu.edu/

Supported Visualization Platforms

Our primary supported visualization platforms are Linux, IRIX, and Windows; however, we can also provide limited support for Mac OS X, Solaris, and AIX. Whenever possible, we attempt to identify and adopt tools that are platform independent; however, some special-purpose displays, hardware cards, and software packages may impose certain platform restrictions.

Visualization Services

UITS supports the following general services in the area of visualization:

- **Software Consulting** – By consulting with our visualization staff, we may be able to help you derive more functionality out of the applications that you are currently using, or we may be able to help you identify a better application or suite of tools for your needs. We can also assist with data translation tasks as well as application customization through custom scripts, macros, or plug-ins.
- **Custom Software Development** – In some situations, there may be no software tools satisfactory for your visualization task. In such cases we may be able to partner with you and your staff to develop a custom application that precisely fits your needs. While such developments may take several months or more, the benefits of having an application custom designed to meet your needs can be extremely valuable. We have developed custom applications for volume rendering, molecular visualization, and phylogenetic tree viewing in the past, and are currently developing a scalable pedigree tree visualization system.
- **Hardware Facilities** – While many visualization tasks can be carried out on desktop workstations, some tasks may have highly specialized hardware or display requirements, or may have computation or memory requirements that exceed those of your desktop system. Through the facilities of the UITS Advanced Visualization Lab in SL 239, IU researchers can experience and utilize a range of advanced visualization systems, including large-format stereo projection systems, very high resolution displays, a force-feedback device, a distributed rendering cluster, special-purpose volume-rendering hardware, and a selection of tele-conferencing and tele-collaboration systems.
- **Hardware Consulting** – While central visualization facilities are satisfactory for occasional use, once-off demonstrations, or technology "test drives", we realize that visualization technologies have the greatest impact on the scientific process when they are conveniently and directly accessible in the researcher's lab or

department. Our staff can help you to derive specifications for the system or systems that best fit your needs, including desktop workstations, specialized hardware, and advanced displays.

Specific Hardware Facilities

The following systems are available for demonstration, testing, and routine use in the AVL facility in SL 239 on the IUPUI campus:

- **Large-format stereo display** – The ImmersaDesk provides a semi-immersive visualization experience with a 4'x5' stereoscopic screen and full spatial tracking. It is ideal for interactive data exploration, education experiences, and small group collaborations. The AVL is also developing a new portable, passive-stereo display that incorporates many of the features of the IDesk, but with easier maintenance, higher luminosity, and a significantly lower cost.
- **Reachin haptic display** – This system couples a Phantom haptic (force-feedback) device with a stereo monitor and a half-silvered mirror to co-locate the graphical rendering with the haptic rendering. Haptic output provides a powerful complement to graphical output, and can be applicable to volume exploration, molecular simulation, and multi-dimensional data analysis.
- **VolumePro** – Terarecon's PC-based VolumePro cards provide hardware implementations of specific volume rendering algorithms. They provide features that are not attainable on other systems, including real-time performance on large data sets as well as interactive lighting and transfer function editing. The AVL has trailed the original VP500 and is in the process of evaluating the new VP1000.
- **Very high resolution LCD** – The IBM T221 ("Bertha") display is a 22.2" LCD panel with 3840x2400 resolution yielding 204 DPI resolution. Connected to a single PC with a high-end graphics card, it is compatible with most existing desktop software and is proving quite valuable for a variety of imaging and information visualization tasks. It has been tested thoroughly under Windows, and Linux testing is underway.
- **Rendering Cluster** – There is a growing number of visualization tasks and data sets that exceed the rendering capabilities of a single workstation. For many of these problems, it is possible to use clusters of workstations to render and display the data in parallel. Although this technology is still relatively new and experimental, the AVL does offer a small eight-way cluster for testing different topologies and software distribution tools.
- **Tele-conferencing systems** – The Polycom Viewstation provides high-fidelity video and voice over IP for point-to-point tele-conferencing, or pre-arranged multipoint conferencing through a central MCU. The Access Grid Node is an alternative PC-based technology that uses multicast networking. Each of these videoconferencing technologies can be used in conjunction with application-

sharing, file-sharing, and white-boarding tools such as VNC and NetMeeting to create an effective collaborative workspace for small groups of collaborators.

Specific Supported Software

- **General-purpose Visualization Software** – We have conducted extensive investigations into OpenDX and VTK as standard frameworks for visualization development.
  - OpenDX provides great flexibility for network distribution and multi-platform interoperability as either an end-user application or as a high-level development tool. Learn more at:

    http://www.opendx.org/
  - VTK provides standard visualization algorithms within a traditional programming environment, and includes extensions to support VolumePro hardware as well as versions which permit distributed parallel rendering on PC clusters. Learn more at:

    http://www.kitware.com/vtk/
- **Specific-purpose & Custom Software** – In addition to general-purpose software tools, we support a variety of specialized open-source visualization tools, as well as several locally-developed codes, including the following:
  - **3DIVE** (3D Interactive Volume Explorer) is a visualization tool for interactively displaying and manipulating volumetric data. Developed by the AVL in conjunction with the IUPUI Computer Science Department and groups within the IU School of Medicine, 3DIVE runs on the ImmersaDesk and other SGI systems and is currently being ported to the Linux platform and other display devices.
  - **XMView** (X-Windows Molecular Viewer) is a multi-platform software tool developed by the AVL in conjunction with the IU Molecular Structure Center. Originally targeted for use with small, crystalline structures, it is being extended to support macromolecular visualization techniques and file formats of interest to biochemists.

XIII. For additional help

Help with Windows or Mac OS

For help with your Windows or Macintosh computer, including installing and using an SSH client, contact the Support Center by phone. At IUPUI dial (317) 274-4357. At IUB dial (812) 855-6789. You can also search the Knowledge Base at:

http://kb.iu.edu/

Parallel algorithm/program development

For help with parallel algorithm development, parallel programming, and getting your program running correctly and efficiently, contact High Performance Computing at *hpc@indiana.edu*. Visit the HPC Web site at:

http://www.indiana.edu/~rac/hpc/index.shtml

Biological computing

For help in biological computing, particularly in the areas of genomics, cell biology, and molecular biology, contact Bioinformatics Support at *bioinfor@iu.edu*. Visit the BIOS Web site at:

http://www.indiana.edu/~rac/bioinformatics/

Statistical and mathematical software

For help with statistical and mathematical software such as SPSS, contact the Stat/Math Center at *statmath@indiana.edu*. Visit the Stat/Math Web site at:

http://www.indiana.edu/~statmath/

Advanced visualization

For help with visualization, or to schedule a consultation or general technology demonstration, contact the Advanced Visualization Lab at *avl@avl.iu.edu* or call 856-4911. Visit the AVL Web site at:

http://www.avl.iu.edu/

Data storage

For help with your massive data storage requirements contact the Distributed Storage Services Group at *store-admin@iu.edu*. Visit the DSSG Web site at:

http://www.indiana.edu/~dssg/

System administration

For general supercomputer system administration help contact Research and Technical Services at *rats@indiana.edu*. Visit the RATS Web site at:

http://www.indiana.edu/~rats/

More information on the Research SP

The Research SP home page at http://sp-www.iu.edu/ contains a wealth of information. Documents on that page especially helpful for beginners are:

- Getting Started on the Research SP:

  http://sp-www.iu.edu/getting.started.shtml

- How to Change Your Password or Login Shell:

  http://sp-www.iu.edu/passwd.shtml

- Submitting Batch Jobs to LoadLeveler:

  http://sp-www.iu.edu/LLguide.shtml

- LoadLeveler Configuration:

  http://sp-www.iu.edu/LL.config.shtml

- LoadLeveler and Maui User Commands:

  http://sp-www.iu.edu/LL.commands.shtml

- Software Available on the Research SP:

  http://sp-www.iu.edu/SP.software.shtml

More information on Unix

- A complete set of Unix tutorials authored by the Unix Systems Support Group (USSG):

  http://www.ussg.indiana.edu/uhelp/tutorials/toc.html

- Information on the STEPS/PROSTEPS classes about Unix:

  http://ittraining.iu.edu/

- The Stat/Math Center's guides to Unix and to using statistical and math software under Unix:

  http://www.indiana.edu/~statmath/support/byos/unix/index.html

- More information on Fortran/C/C++, SPSS/SAS, and other stat/math software:

  http://sp-www.iu.edu/xlf7.1/index.htm

- Reference manual for IBM C:

  http://sp-www.iu.edu/VAC++6.0.shtml

- Tutorials and documentation on SPSS, SAS, and other statistical and mathematical software:

  http://www.indiana.edu/~statmath/support/bytitle/index.html

- A course on the tools of scientific computing within a Unix environment:

  http://beige.ucs.indiana.edu/P573/

Information on parallel programming

Although you should contact HPC at *hpc@indiana.edu* for assistance with your parallel computing projects, you may benefit by working through the online tutorials at:

http://www.iu.edu/~rac/hpc/mpi_tutorial/index.html

http://www.cs.indiana.edu/classes/b673/notes/mpi1.html

http://beige.ucs.indiana.edu/B673/node113.html

Acknowledgments

Cover illustrations (top and bottom) based on original photographs © Tygan Miller. Middle illustration courtesy Indiana University's Advanced Visualization Laboratory.

The facilities described in this document were made possible in part through funding from Indiana University, the Indiana University Office of the Vice President for Information Technology, the State of Indiana, Shared University Research Grants from IBM, Inc., the National Science Foundation under Grant No. 0116050 and Grant CDA-9601632, and from the Lilly Endowment through their support of the Indiana Genomics Initiative. The Indiana Genomics Initiative (INGEN) of Indiana University is supported in part by Lilly Endowment Inc.

Contributing authors

Robert Cruise
David Hart
Mary Papakhian
Richard Repasky
John Samuel
Anurag Shankar
Craig Stewart
Eric Wernert

Editor

Malinda Lingwall