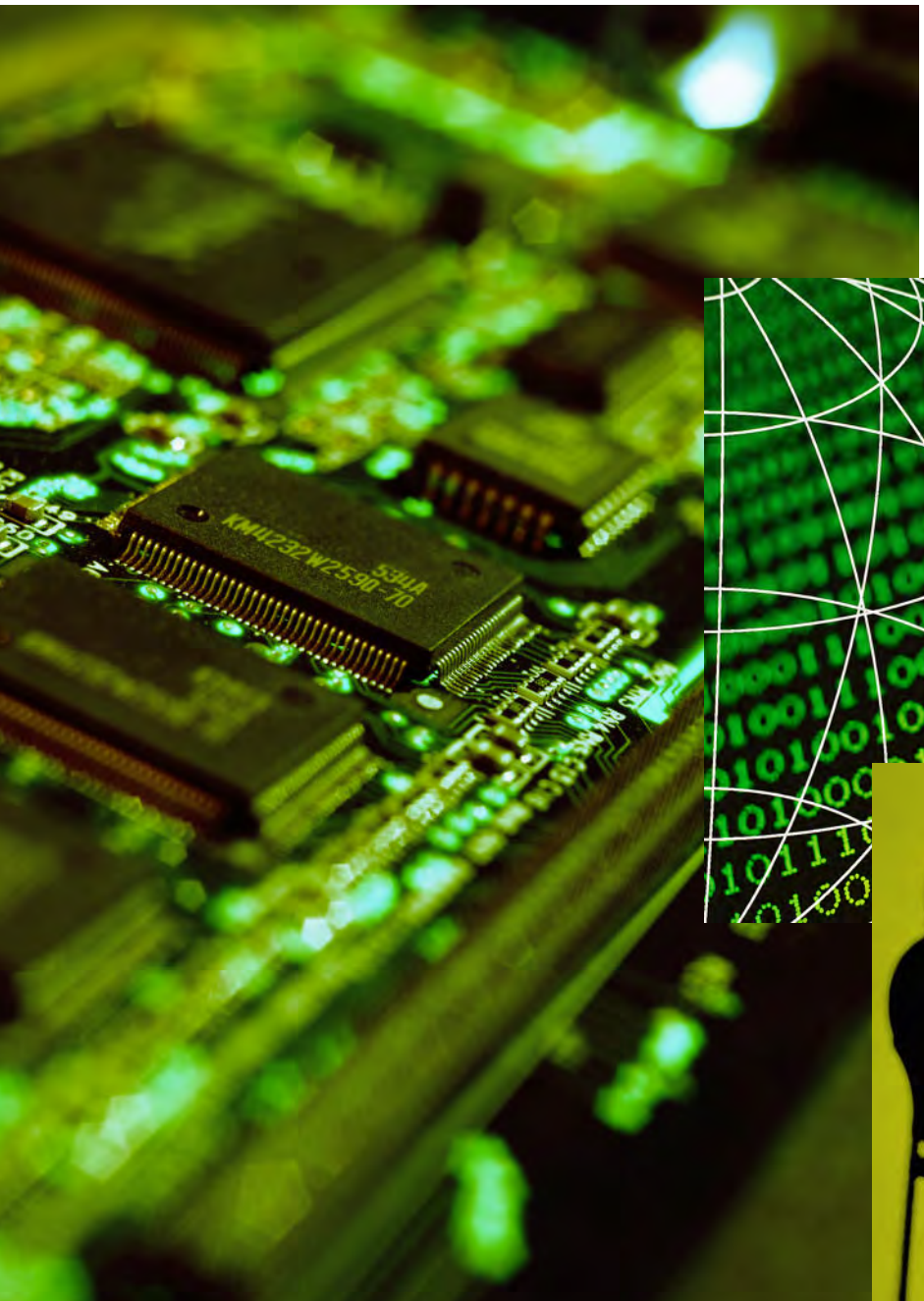
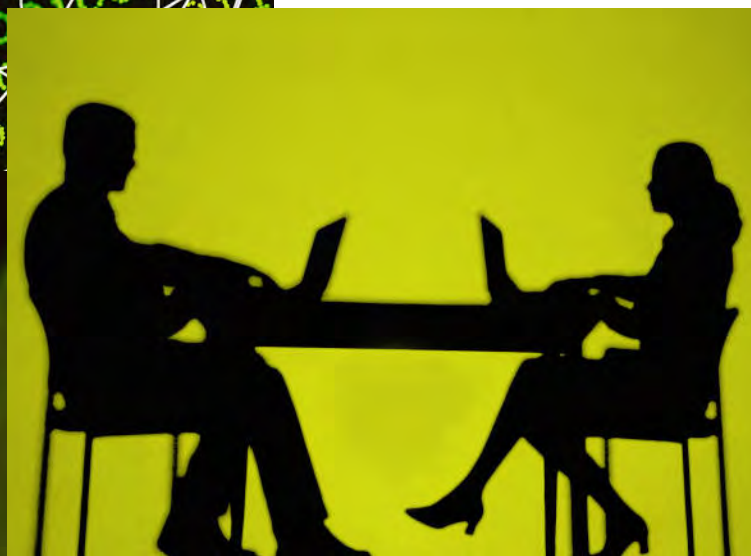
Indiana University's Advanced Cyberinfrastructure



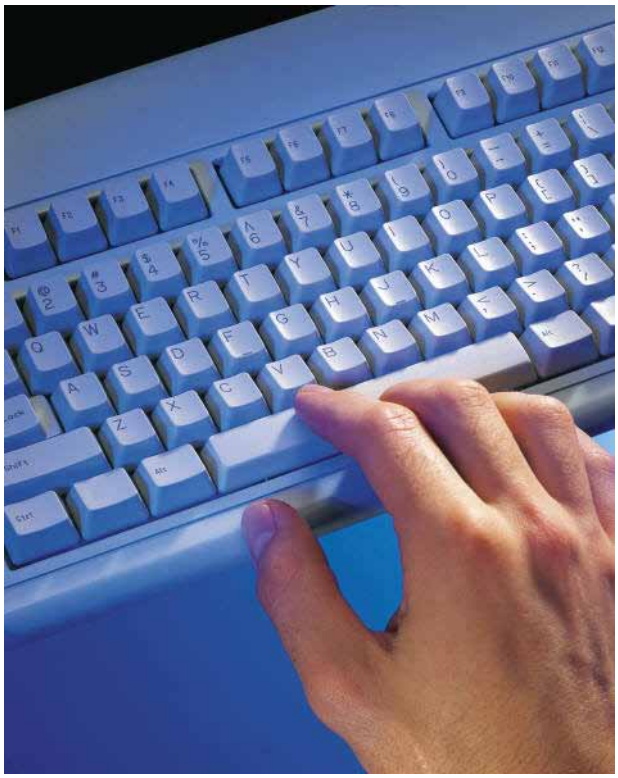


Version 1.0
Revised August 2011

Indiana University's Advanced Cyberinfrastructure

Version 1.0
Revised August 2011

Contents



Preface

For most researchers, information technology is a tool – a means, not an end. This is reflected in the term cyberinfrastructure, which Indiana University (IU) defines as high performance computers, massive data storage systems, data resources, advanced instruments, sensor networks, and people, linked together by advanced software and high performance networks to improve research productivity and enable breakthroughs not otherwise possible. Ideally, researchers should think about computing power and data storage in no more detail than is given to thinking about the generation of electrical power that feeds the outlets in a lab. While computer scientists and IT researchers are striving toward this goal, we're not there yet.

The purpose of this document is to introduce researchers to Indiana University's cyberinfrastructure – to clarify what these facilities make possible, to discuss how to use them and the professional staff available to work with you. The resources described here are complex and varied, among the most advanced in the world.

The intended audience is anyone unfamiliar with IU's cyberinfrastructure. Many readers will find sections (for example, introducing Unix) with which they are familiar and want to skip. In some cases there is information that does not need to be learned, but that will help with a one-time process: logging into a particular computer for the first time, for example, involves steps that may be followed keystroke-by-keystroke and then forgotten.

This document is a starting point, not a comprehensive guide. Its intent is to get you off to a good start, and then point you in the direction of online resources and UITS consulting staff. It is provided for the convenience of researchers, to permit anyone at all familiar with computers and the Internet to get a general overview of IU's cyberinfrastructure and what it offers.

1. Computing at IU

1.1. Introduction

The cyberinfrastructure at Indiana University is among the best at any university in the world. Its supercomputers, data storage systems, visualization environments, and access to high performance research networks are intended to help you, the researcher, achieve breakthroughs in your scholarship by making possible new calculations, analyses, and visualizations of massive amounts of data.

This document is intended to help you get started using IU's cyberinfrastructure for your research.

To see how these systems can help you, ask yourself:

- What am I doing with computers now that I would like to do faster, in more detail, or on a larger scale?
- What kind of research questions would I tackle, if computing resources and expertise were not a limiting factor?

Most services provided by the Research Technologies division of University Information Technology Services (UITS) are baseline services, offered with no direct charge to the user, as long as the purpose is research or education. UITS provides hardware, software, networks, and support to enhance research and instruction.

This document focuses only on research technologies, and is not a comprehensive guide to the IU information technology environment. General purpose computing (for example, email) is important, but a document describing *all* of UITS' services would be very long and difficult to parse. However, the first section of this document does include pointers to information about other services. For a full description of UITS services, see the UITS Services and Support page online at:

http://uits.iu.edu/

For a description of services available through Research Technologies (a division of UITS and affiliated with the Pervasive Technologies Institute), see their homepage online at:

http://pti.iu.edu/rt/

If you have questions, email to researchtechnologies@iu.edu is always welcome.

TYPOGRAPHIC CONVENTIONS USED IN THIS DOCUMENT

We will present examples as follows:

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

`Text that you are to type verbatim is boldfaced.`

`Text that may change, such as a name, is italicized.`

Clickable hyperlinks and email addresses are underlined.

Combination keystrokes are joined with a slash (e.g., CTRL/z means hold down the "CTRL" key while you press the "z" key. Sometimes ^Z is also used to indicate this.

When you are instructed to *enter* some text, do so, and then press the RETURN or ENTER key (depending on your keyboard). When you are instructed to *press* a key, press that key alone; do not press RETURN or ENTER.

1.2. Computing support at IU

Support for computing at IU is provided through a tiered system, starting with departmental local support providers, who set up Windows and Macintosh workstations, and connect them to the campus network.

Basic computing information is available from the UITS Support Center and the IU Knowledge Base. Help from the IUB and IUPUI Support Centers is available 24/7 at:

IU Bloomington:	812-855-6789
IUPUI:	317-274-HELP (4357)
Email:	ithelp@iu.edu
Web:	https://kb.iu.edu/

For Support Center contact information at all IU campuses see "How can I contact the Support Center at each IU Campus for help?" at:

http://kb.iu.edu/data/abxl.html

The Support Center provides excellent support for Windows, Macintosh, Linux, and Unix computing, but they don't provide in-depth support for research systems.

The IU Knowledge Base (KB) is more than just Frequently Asked Questions. For UITS and Research Technologies, the KB is the primary means for delivering documentation. It contains more than 7,000 answers to questions about information technology at IU and in general. The KB includes a full-text search that returns documents matching your search terms (most efficient for experienced users), menus that sort common questions by topic (easiest for beginners), and a glossary that defines important terms:

Search:	https://kb.iu.edu/
Menus:	https://kb.iu.edu/data/menu.html
Glossary:	https://kb.iu.edu/data/glos.html

This document assumes you have a personal computer from which you can access UITS research computing, data storage, and visualization systems. Beginners sometimes ask about gaining physical access to IU's computing facilities, but at IU you don't need to travel to a "Computing Center." Instead, you can access UITS research computing systems through the network, using your desktop computer, a laptop, or even a mobile device. There is no need to ever visit (or even know the location of) a supercomputer or storage disk farm. And, it's not likely you'll notice any difference between using a system located in Bloomington and one located in Indianapolis.

This document also assumes you already have your basic IU computer and email account (your Network ID, or just NetID). If you don't, see "How do I get my first computing accounts at IU?" in the IU KB:

http://kb.iu.edu/data/achn.html

2. Research Technologies

2.1. Advanced IT services

As with UITS services generally, advanced IT services are available at IU without charge for research and instruction. The Research Technologies division's services are focused on a few main areas:

Supercomputing – By using IU's supercomputing facilities, researchers can drastically accelerate their simulations and analyses, performing massive new computations that are not otherwise practical.

IU has two supercomputers: Big Red and Quarry. Each is built from many processors, which are organized into nodes. More information on these systems is included in later sections of this document, and online. The platform or platforms you choose to use will be a matter of the requirements of your programs in terms of operating system, software packages, computing power, and available memory, as well as your own personal preferences. You can ask for advice.

Data services – IU's data storage facilities make possible the secure and reliable storage of massive amounts of data. Anyone who uses more data than can fit on a single CD can benefit from using these systems. And anyone who has ever said, "I'd like to pursue a research idea, but can't store that much data" will most likely find that storage is no longer a limiting factor. In addition, UITS supports use of research databases and publishing data via the web.

Consulting – Research Technologies provides consulting and support for:

- Scientific visualization, virtual reality, and high-end computer graphics
- Statistical, mathematical, and bioinformatics software
- Data storage and database design
- Programming supercomputers

This is *not* an exhaustive list.

2.2. Why use supercomputers?

Supercomputers (also called *high performance computers*) make it possible to run applications more quickly and to perform computations that would otherwise take unreasonably long. For example, Research Technologies staff worked with one School of Medicine researcher who uses a program that typically took a week to analyze a single data set on the computer in his lab. The program now finishes in an hour.

There are three main ways supercomputers make programs run faster: better single-processor performance, trivially parallel processing, and parallel programming.

2.2.1. Single-processor performance

Programs often run faster on a supercomputer than on other computers for three reasons:

- Due to the chip ***design***, a combination of speed, caching, and other optimizations in the hardware, as well as optimizations of software, numerically intensive tasks often run faster on supercomputers than on personal computers.
- The supercomputer has more ***memory*** than most computers, so it can hold more data in memory at a given time. Your personal computer might have to write pieces of intermediate calculations out to its hard disk due to limited memory.
- A supercomputer may have better hard ***disk performance*** than most desktop systems. There are a number of features engineered into IU's supercomputers that make transferring data to and from the hard disk very fast.

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, you may be unfamiliar with the environment provided by supercomputers. This document should help you over this "threshold" by familiarizing you with the Unix environment.

Also, many commercial programs that run on Microsoft or Apple operating systems (e.g., SAS and MATLAB) also run on Unix in exactly the same way.

Finally, if you have programs that you or a colleague have written for other Unix systems, modifying these programs to run on IU's supercomputers (i.e., "porting" the code) is typically very easy (and something with which Research Technologies staff will gladly help). Programs written for Microsoft DOS or Windows, or Apple Macintosh, can usually be ported as well. The simplest method for speeding up your computing may be to take the program you are currently running on a personal or a departmental computer, and with the help of UITS staff, get it to run on one of IU's supercomputers.

2.2.2. Trivially parallel processing

Suppose you have a program that runs on a single processor, and you have many different data sets to analyze. Thanks to the many processors of IU's supercomputers, you can run the same program dozens of times simultaneously, each time analyzing a different data set. This could be done by repeatedly submitting jobs with varying input or parameters, but the process can usually be automated using simple programs called "shell scripts." This is the most common and most easily mastered form of "parallel computing."

Trivially parallel processing is a computer science term – "perfectly parallel processing" might be more descriptive. This is a very straightforward way to achieve tremendous speedups in your data analysis. Research Technologies staff will help you implement the scripts necessary to do this sort of parallel processing, usually in a matter of days.

2.2.3. Not-so-trivial parallel programming

It is not always a simple matter to organize the work a program does so you can cut the time required in half by doubling the number of processors working on it. In fact, that is the theoretical limit of what can be done, and this limit is rarely achieved.

This is complicated work requiring expertise, and it can take weeks or months to convert from running on a single processor to running efficiently on multiple processors simultaneously. Parallel programming is not simple, but it may be the key to speeding up your applications. Research Technologies staff members are available to work with you to parallelize your programs so you can finish your work more quickly.

2.3. Data services

2.3.1. Data storage and management

Researchers today may deal with all sorts of computer data – from field research, instruments, surveys, and computer simulations. A lot of data ultimately end up on CDs, or on the hard disks of personal computers and departmental servers. The time, discipline, and hardware costs required to back up this data can be formidable, and often means that, in practice, no backups are available. Consider the following situations:

- You are unable to take on a project because you do not have enough storage.
- You have an instrument in your lab producing data at a rate or a volume you cannot handle.
- You want to access your data securely and transparently from anywhere.
- You need a place to store your data while you are out of the country.
- You want truly reliable backups.

If any of these apply, Research Technologies can help. We won't back up your data for you, but we'll provide a safe and convenient place for you to do so.

You are likely familiar with storing data on CDs or DVDs, which hold about 700 MB or 4.7GB of data, respectively. Some instruments are set to write the results of a particular analysis onto a disk, which is then handed to the researcher. Using such a method, data retrieval can be agonizingly slow. UITS provides specialized systems capable of capturing data from modern digital instruments, and accessing it again at GB/sec rates. The storage capacity available is world-class. More details are given in "Archival and long term storage" on page 29.

2.3.2. Databases

The Research Database Complex (RDC) is configured to support both Oracle and MySQL databases. If you have data you'd like to store, query, and retrieve easily, UITS can set up a database on the RDC. The host, disk space, database management software, backups and patches are all provided. These databases can be configured to restrict access to only those people you specify. Use of these central systems gives you the advantage of extremely good database performance, database administration provided by UITS, and professionally managed backups.

2.4. Consulting and support

The Research Technologies division can be split roughly into two groups: one running the hardware and the other supporting research applications. For the latter, consulting is paramount:

- The staff of the UITS **Advanced Visualization Laboratory** (AVL) is available to help you with the latest visualization techniques and technologies. Visualization allows researchers to analyze their data in whole and in detail, and to rapidly spot trends, recognize relationships, and identify anomalies. Specific services and facilities offered are detailed later in this document, in the section "Using IU's visualization facilities."
  http://pti.iu.edu/avl
- **Bioinformatics Support** provides help with biological computing, particularly in the areas of genomics, cell biology, and molecular biology.
  http://kb.iu.edu/data/alei.html
- The **Biomedical Applications** group provides consulting and custom development to help biomedical researchers access and manage their data.
  http://pti.iu.edu/rtl/biomed-apps-data-collections
- The **Center for Statistical and Mathematical Computing** provides help with statistical and mathematical software (e.g., SAS, Mathematica, and MATLAB), on personal computers, as well as supercomputers. If you'd like this software for your department or personal computer, very favorable licensing arrangements are available.
  http://www.indiana.edu/~statmath
- The **Digital Library Program** (DLP), a joint effort of the IU Libraries and UITS, provides consulting and other services related to digital library development.
  http://www.dlib.indiana.edu
- The **High Performance Applications** (HPA) group provides programming support, and can answer questions about how you can gain the advantages supercomputers can offer. They will help you to migrate, optimize, and parallelize your code, and they manage licenses for programming tools, such as compilers, libraries, debuggers, and performance analyzers. They also will help you use the resources of the TeraGrid.
  http://pti.iu.edu/hpa
- Research Technologies **Core Services** provides enterprise Linux licensing (RedHat, SUSE), plus public access to free/open source software, a source code repository/version control system, web application hosting on the Research Database Complex, and RenderPortal, a Condor-based rendering service.
  http://pti.iu.edu/cs

- The **High Performance Systems** (HPS) group provides high-performance computing systems that are dedicated to research. These systems are configured to optimize stability and throughput for CPU intensive, I/O intensive, and/or memory intensive jobs. The HPS group supports Indiana University's teaching and research mission by providing and facilitating the use of these centralized research computing resources.
  http://pti.iu.edu/hps
- The **Research Storage** group administers and supports the Scholarly Data Archive (SDA) and the Research File System (RFS). These two systems provide storage services for Indiana University researchers. The Scholarly Data Archive is primarily a tape based archive storage system and is geographically distributed between the IUPUI & IUB campuses. RFS can be mounted on the desktop or accessed over the web or SFTP protocol. RFS supports active editing of files and documents unlike the SDA.
  http://pti.iu.edu/rs

These groups are interested in long-term projects offering opportunities to acquire expertise that can be made broadly available to the university community. They include many PhDs, and frequently participate in externally funded projects.

2.5. Related services

2.5.1. Slashtmp

Slashtmp is a service that lets IU graduate students, faculty, and staff share files via a web interface. It's especially helpful for sharing files that are too large for email. Detailed instructions are available on the Slashtmp site:

https://www.slashtmp.iu.edu/

To use this service you'll be asked to enter your Network ID and password.

2.5.2. Grid computing

A "grid" is a way to combine supercomputers that are in multiple locations and managed by multiple organizations. IU is part of two national computing projects, the TeraGrid and the Open Science Grid. Both are multi-year efforts to build and deploy the world's largest and most comprehensive distributed infrastructure for unclassified scientific research. These grids provide teraflops of computing power, facilities capable of managing and storing several petabytes of data, high-resolution visualization environments, scientific instruments, large datasets, and computing portals and toolkits. Their components are tightly integrated through high capacity networks. Detailed information on grid computing is beyond the scope of this document. The Research Technologies division provides support for TeraGrid users, as discussed below.

For more information about the Open Science Grid, a TeraGrid Science Gateway, see:

http://www.opensciencegrid.org/

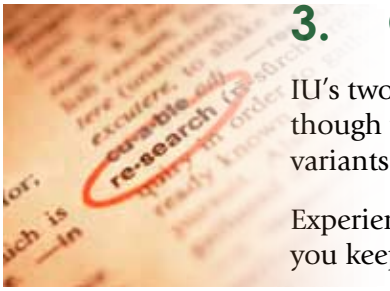
In the summer of 2011 the National Science Foundation Office of Cyberinfrastructure's eXtreme Digital (XD) program began. This program will replace and expand upon the existing TeraGrid program. XD will provide the research community with significantly enhanced high-performance computing, visualization, and data services. Some TeraGrid systems will retire during this transition process, and new ones will be introduced in the future. See the resource catalog for the most up-to-date information on these systems:

https://www.teragrid.org/web/user-support/compute_resources

2.5.3. Videoconferencing

IU offers several videoconferencing services, including administrative group videoconferencing, desktop videoconferencing, and streaming media. UITS Digital Media Network Services (DMNS) – not a part of Research Technologies – has evaluated a variety of products for university-based desktop video conferencing. For more, see the DMNS web site:

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



3. Getting started with IU's research computers

IU's two supercomputers, Big Red and Quarry, appear mostly the same to users even though their compilers and job management systems vary. IU's research computers all run variants of the Linux or Unix operating system.

Experienced users may find most of this chapter routine, and can skip to "Where should you keep your data?" on page 20.

3.1. Requesting research accounts at IU

To obtain an account on Quarry, Big Red, or the Research Database Complex:

1. Use the UITS Account Management Service: https://itaccounts.iu.edu/
2. Select `Manage my IU computing accounts`.
3. When you are prompted to log into the Central Authentication Service, enter your Network ID username and passphrase.
4. Click `create more accounts`.
5. Use the radio buttons to select the accounts you want to create, and then click `Create Account`.

As mentioned previously, these services are available without charge for research and education purposes. You only need to:

- Acknowledge use of IU's cyberinfrastructure in your publications
- Respond to occasional requests for citations, so IU can measure the impact its cyberinfrastructure is having on scholarly activity.

3.2. Connecting to research systems at IU

The Research Technologies division of UITS provides robust, reliable services to the IU community. A critical part of making any computer system reliable is keeping it secure. To protect UITS research systems, all sessions must be encrypted. The program SSH (Secure Shell) is a secure replacement for telnet and the Berkeley r-utilities (rlogin, rsh, rcp, and rdist). It provides an encrypted channel for logging into another computer over a network, executing commands on a remote computer, and moving files from one computer to another. SSH provides strong host-to-host and user authentication, as well as secure encrypted communications over a non-secure Internet.

SSH2 is a more secure, efficient, and portable version of SSH that includes SFTP, which is functionally similar to FTP, but is SSH2 encrypted. At IU, UITS has upgraded its central systems to SSH2 (usually the OpenSSH version), and encourages those concerned with secure communications to connect using SSH2 clients.

Mac OS X comes with OpenSSH built in. For Windows, you will have to download a third-party SSH client, such as PuTTY or WinSCP (see "Logging in from Windows" below).

Note: SSH Secure Shell Client for Windows is no longer available for download from IUware or SSH Communications.

The most common source of confusion for a new user of Unix systems is when there are programs running on your personal computer and programs running on a remote computer, and both are making things happen, and you can interact with either! This can affect cut and paste editing operations, for example, and is discussed further in "Editing files" on page 13 and "What to do when confusing things happen" on page 16.

3.3. Getting started with Quarry

3.3.1. Logging in from Windows

In spring 2008, SSH Communications Security discontinued support for non-commercial versions of SSH Secure Shell, therefore SSH Secure Shell is no longer available for download from IUware or the SSH web site. For more, see SSH's Non-commercial Downloads page:

http://www.ssh.com/support/downloads/secureshellwks/non-commercial.html

At IU, the University Information Security Office (UISO) recommends the following alternatives. IU students, faculty, staff members can download the software for free from IUware using the links provided:

- PuTTY (command line SSH interface): http://iuware.iu.edu/title.aspx?id=781
- WinSCP (GUI client for transferring files): http://iuware.iu.edu/title.aspx?id=785

For more on these applications, see their respective home pages:

- PuTTY: http://www.chiark.greenend.org.uk/~sgtatham/putty/
- WinSCP: http://winscp.net/eng/index.php

To log into a research system from a Windows workstation:

1. Double-click the PuTTY or WinSCP icon on your computer.
2. When the program opens, enter the host name (**`quarry.uits.indiana.edu`**) and your Network ID username.
3. Leave the port at its default value (22), and then select the authentication method `Password`.
4. The first time you connect to this system, a window will appear asking if you want to save the host key. Select `Yes`. Another window will appear, requesting your password. Enter your Network ID passphrase.
5. You will be offered the opportunity to save a new profile. Enter **`Quarry`**.

After the first time, you can select the Profiles menu, select `Quarry`, and go straight to entering your passphrase.

3.3.2. Logging in from Macintosh or Linux

At IU, UITS supports and recommends these clients for secure login:

- Cyberduck, a donationware program, is available for Mac OS X 10.4 and later. IU students, faculty, and staff can download Cyberduck from IUware: http://iuware.iu.edu/title.aspx?id=857
- OpenSSH, the command line SSH suite for Unix, is available from the OpenSSH web site: http://www.openssh.org/
- OpenSSH is packaged with Mac OS X, and is accessible through the Terminal. For instructions, see "In Mac OS X, how do I use OpenSSH?" in the IU KB: http://kb.iu.edu/data/aiag.html

If your personal workstation is running Linux or Mac OS X, log into a research system by opening a terminal window, and at the prompt, enter:

```
ssh quarry.uits.indiana.edu
```

When prompted for your password, enter your Network ID passphrase (you won't see anything when you're typing your passphrase, or you might see asterisks instead of the characters you're typing). Next, you will be greeted with the Message of the Day (*read it, please!*).

3.3.3. First time login

When you use a computer running Unix, you interact with a piece of software known as the *shell*. The shell is text oriented – it was designed before the computer mouse was available.

Note: The Unix shell, which you use to interact with the Unix operating system, is different from the Secure Shell, which you use to connect your computer to a remote Unix system.

There is more than one shell program available. The very first time you log into a UITS supercomputer, the program **changeshell** is executed, which asks you to select your preferred login shell. Unless you are an experienced Unix user and have a specific reason to do otherwise, **we strongly recommend selecting the bash shell**. The default shell on Quarry is bash. The first time you log in, you'll see the following greeting:

```
This program will assist you in changing your login
shell on all nodes of the Quarry cluster.

1) bash
2) tcsh
. . .
5) quit

Select 1-5:
```

Enter **1**. You will see:

```
Changing shell for username.
Password:

Shell changed.
Your shell has been changed to the bash shell.
This will take effect on all nodes within 15 minutes

Select 1-5:
```

Type **5**, and then press Enter to exit the system. You will receive the message:

```
Connection to quarry.uits.indiana.edu closed.
```

Once you select your preferred login shell, it will be set automatically on the node you're currently on. However, it may take up to 15 minutes for it to be propagated to all nodes.

Each time you login thereafter, you will see the Message of the Day, and then the Unix prompt.

To verify your shell selection, log into Quarry again, and at the prompt, enter:

```
echo $SHELL
```

If you selected the default shell, bash, you will see:

```
/bin/bash
```

3.3.4. Logging out

To end your Unix session, enter:

```
logout
```

The system may respond with the following:

```
There are suspended jobs.
```

If so, simply repeat the **logout** command. **Always log out** when you are finished using a system. Never leave an active session unattended!

3.4. Getting started with Big Red

Note: Big Red was retired from service on the TeraGrid July 31, 2011. Access via SSH keys or GSI-SSH for TeraGrid users will remain enabled for another three months. All data stored in TeraGrid accounts on Big Red should be backed up prior to October 31, 2011.

To obtain an account on Big Red, IU researchers should use the Account Management Service:

https://itaccounts.iu.edu/

3.4.1. First time log in

To log in to Big Red, follow the instructions presented in "Getting started with Quarry" on page 10, substituting `bigred.teragrid.iu.edu` **for** `quarry.uits.indiana.edu`.

Its inclusion in the TeraGrid makes Big Red is a national resource. Big Red is dedicated primarily to large-scale analyses using parallel programs. This document makes no attempt to introduce parallel programming – various academic departments have courses for that.

Note: If you want your usage on Big Red to be charged to a TeraGrid account, you should log in using the hostname:

```
login.bigred.iu.teragrid.org
```

Logging into Big Red for the first time brings up the firstshell script, just like on Quarry. You'll see the following greeting:

```
Welcome to Big Red!

This program is run the very first time you log in
to Big Red to allow you to select your 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:
```

Unless you're an experienced Unix user and have a specific reason to do otherwise, UITS strongly recommends selecting **bash**. Enter **4**, and then press Enter. Your change will be confirmed:

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

Type **6**, and then press Enter to exit the system. You'll see:

```
Connection to bigred.teragrid.iu.edu closed.
```

Once you select your preferred login shell, it will be immediately set on the node you were logged into, but it might take up to 60 minutes for the setting to propagate to every node on Big Red.

Afterward, each time you log into Big Red, you'll see the login banner, which displays current announcements and information regarding the status of the cluster.

4. Basic interaction with Unix operating systems

Unix has a powerful environment designed by and for computer programmers. Beginners and casual users may become frustrated by the command line interface and the jargon-filled help system, but most users spend little time on the Unix command line, working instead within applications like SAS. This document will show you how to work with the Unix command line and access the applications you need.

When you log into an interactive session, the Unix system runs a shell program that puts a prompt on your terminal screen and processes your commands. Prompts may vary (some common ones are `$`, `%`, and `>`). This document uses `$` to represent the Unix system.

To run a program, enter its name at the command prompt:

```
$ hostname          [The shell program supplied the $; you enter hostname.]
b003                [This is the system's response; it may vary.]
```

The `$` character is the prompt. It indicates the shell is ready for a command. (After this section, the Unix prompt will no longer be included in the examples.) The system's response shows the user is connected to the "003" node of Quarry (you may get a different node – a node is one of the parts of the system, and is of no particular interest in itself).

Unix is case sensitive. When you type a command, you must use uppercase or lowercase letters as they appear in this document. Most Unix commands are lowercase only.

An online manual, known as man (short for manual) pages, is available to help you learn about particular Unix commands and functions. To learn about a command or function, read its man page:

1. At the Unix prompt, type **`man`** and the name of the command (e.g., **`man date`**), and then press Enter.
2. The system will return a man page for the `date` command, which will include a description of the command or function, including any command line options or parameters, and examples of common usage.
3. To see additional pages, press the Spacebar.
4. To exit a man page, type *`q`*.

4.1. Editing files

On Quarry, to create new files or edit existing ones, we recommend using the editing program Nano. To use Nano to edit the file "note-to-myself.txt," at the prompt, enter:

```
nano note-to-myself.txt
```

If the file exists, Nano will open the file. If the file doesn't already exist, Nano will create a new (empty) file with that name. At the bottom of the screen, Nano displays a command summary. For example, the first entry in the command summary reads:

```
^G      Get Help
```

This means, for help, hold down the CTRL key and press **`g`**.

To add text to your file, begin typing. Use the arrow keys to move the cursor. Here are some basic Nano commands:

`CTRL/x` Exit Nano (you'll get a chance to save your file)

`CTRL/o` Save (output) your file

`CTRL/v` Move forward one screen

`CTRL/y` Move back one screen

CTRL/r At the cursor, insert the contents of another file

CTRL/k Cut the cursor's line of text

CTRL/u Paste the previous cut

CTRL/w Search the file for a text string

4.1.1. A common source of confusion

Your terminal program also offers copy-and-paste functionality, but that's part of your workstation's system, not part of the remote Unix system. Keeping this straight may help you avoid much frustration.

In particular, the cut and paste commands for Nano store the text in memory on Quarry. Your terminal emulator (SSH) can cut and paste too, but it stores the text on your workstation, and most likely uses different keystroke combinations to do it (look for a drop-down "Edit" menu). With your workstation's terminal program, you can highlight text anywhere in the terminal window, and copy (but not cut) it. Then, if you paste the text using the terminal program's keystroke combination, it will send the copied text to the Unix program, and insert it at the cursor's current location, which may not be the spot you intended.

There are graphical user interfaces (GUIs) for Unix, but they are beyond the scope of this document. Your interactions with IU's supercomputers will generally be text-based via a terminal window.

4.2. Files and directories

Once logged in, you will be located in your home directory. A Unix directory is the same as a Windows or Mac OS X folder. To find out where you are (i.e., the path to your current working directory), at the command line, enter:

```
pwd
```

You should see a response similar to:

```
/N/u/username/Quarry
```

The shell refers to the current directory as "." (a dot), and to its parent directory as ".." (two dots). To move from the current directory to its parent directory, enter:

```
cd ..
```

Use "**~**" (a tilde) to refer to your home directory (e.g., `~statmath` refers to statmath's home directory). To move to statmath's home directory, enter:

```
cd ~statmath
```

To move to your home directory from wherever you are, enter:

```
cd ~
```

To create a new directory, use the **mkdir** command. For example, to create a directory named NewCode, enter:

```
mkdir NewCode
```

To move to a subdirectory, (e.g., the newly created NewCode directory), enter:

```
cd NewCode
```

Filenames may contain letters, numbers, underscores, dashes, and dots – **but not spaces!**

Unlike Windows, Unix does use filename suffixes (e.g., .doc) to associate files (e.g., .doc files) with an application (e.g., Word). You may find it convenient to use such filename extensions to keep track of your file types (e.g., you might find it useful to end all your SAS command files with .sas).

To see a list of files and subdirectories in the current directory, enter:

```
ls
```

The resulting list of files and subdirectories will look similar to this:

```
Mail                    myjob.sas               note-to-myself.txt
```

To view the contents of the note-to-myself.txt file, use the **cat** (catenate) command. Enter:

```
cat note-to-myself.txt
```

Warning: Use the **cat** command only with text files. If you use it with a "binary" file, it could cause problems with your terminal. See the next section ("Stumbling blocks") to learn some ways to recover from mistakes.

To view the contents of the `note-to-myself.txt` file one screen at a time, use the more command. Enter:

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

If the file is long enough to fill up more than one screen, press the Spacebar to move to the next screen. To go back one screen, press the "b" key, and to quit, press "q" (press only the letter keys; do not press Enter afterward).

To rename a file, use the **mv** (move) command. For example, if you mistyped a filename, calling it `ob.sas` when you meant `job.sas`, to correct the error, enter:

```
mv ob.sas job.sas
```

As a result, the file is renamed `job.sas`, and there is no longer a file named `ob.sas`.

To copy a file, use the **cp** command. For example, to copy `job.sas` to a file called `job2.sas`, enter:

```
cp job.sas job2.sas
```

As a result, the directory now contains two files with identical content, but different filenames. This is a good way to produce a template that's ready for modification.

To delete a file, use the **rm** (remove) command. For example, to remove `note-to-myself.txt`, enter:

```
rm -i note-to-myself.txt
```

The system will reply:

```
remove note-to-myself.txt?
```

If you're certain you want to remove the file, press the "y" key, or if you change your mind and want to keep the file, press the "n" key. (The "`-i`" added to the **rm** command is the "interactive" option; it makes the system ask you to confirm every deletion.)

4.3. Stumbling blocks

4.3.1. Common error messages

Unix error messages can be frustrating. For example, to list the files in your current working directory, the command is **ls**. Here's an example of what can go wrong:

```
LS
```

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

Remember: Unix is case sensitive. If you have Caps Lock set on your keyboard, you'll have to turn it off if you want to list the contents of your working directory.

Here's another example:

```
splus

splus: Command not found.
```

In this example, `Command not found` indicates the system does not recognize "**splus**." That's because "**Splus**" is the correct spelling.

Often, you'll get an error message if you leave off one or more parameters in the command you're trying to execute. For example:

```
mv

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

When using **mv** (which renames a file), you must provide the original filename (`src`) and the new filename (`target`). The options in the square brackets are optional. For details, read the **mv** manual page:

```
man mv
```

4.3.2. Commands to handle with care

Certain Unix commands can be confusing for beginners, and can put your data at risk if you don't know how to use them correctly. In particular, you should not experiment with the commands **chmod**, **emacs**, **vi**, or **passwd**.

The Unix command to delete a file (**rm**) purges files from the system, and there is no way to recover a file once you delete it with **rm**. Likewise, if you use **cp** to copy a file to an existing file, and end up overwriting the existing file, the change is irrevocable. To help avoid such problems, you should use the inquire option (**-i**) as much as you can. For example, **mv**, **rm**, and **cp** each take the **-i** option:

```
mv -i SomeSillyFile ReallyImportantFile

overwrite ReallyImportantFile?
```

Adding the **-i** option gives you a chance to answer **y** (yes) or **n** (no), and could possibly save you from losing important data.

4.3.3. Files to handle with care

In your home directory, certain configuration files establish your working environment. These "hidden" files (they are not listed when you use **ls**) have filenames that start with a dot, and are commonly referred to as "dot files". Dot files are beyond the scope of this document, but you can learn about some common ones in the IU KB:

https://kb.iu.edu/data/afyy.html

4.3.4. What to do when confusing things happen

Whenever you need help, you can check the IU KB (https://kb.iu.edu/), or call the UITS Support Center (5-6789 at IUB, 4-HELP at IUPUI).

If your Delete and Backspace keys don't seem to work correctly, try using **CTRL/h** instead. This can happen if your local terminal program and the remote shell program are not in sync. Usually, you can fix these keys for the rest of your login session by typing **reset** or **stty sane** at the command prompt.

The **stty** command is short for "set teletype" (which gives you an idea of how long ago Unix was created!). If either **reset** or **stty sane** do not fix the problem, enter **stty erase**, and then press the Backspace or Delete key.

If your typing does not display on the screen at all, you may have accidentally frozen your screen by pressing **CTRL/s**. To unfreeze it, press **CTRL/q**.

If you accidentally type something you don't mean, you might find yourself in a program you don't recognize. Don't panic! Here are some commands to try, in recommended order (press Enter or Return after each):

q

Q

quit

exit

stop

logout

Here are some more (they do not require pressing Enter or Return):

The Escape key (Esc)

CTRL/c

CTRL/d

CTRL/q

CTRL/x

CTRL/]

CTRL/z

When you try to log out, you may get the message: `There are stopped jobs`. Simply type **logout** again to kill those jobs and finish logging out.

If all these commands fail, try closing the terminal window. The Unix operating system will log you out automatically after a few minutes. If you're still stuck, try to get a consultant or your support provider to help you. If no one is available to help, as a last resort, reboot your workstation.

4.4. Printing files

If you're running Windows or Mac OS X on your workstation, the easiest way to print a file is to download it (as described in the next section), open it in either WordPad (Windows) or TextEdit (Mac OS X), and then use that application to print the file.

Sometimes, you can end up with messy output, because the text in the file extends beyond the present margins in your application. As a workaround, switch your application's printing preference to landscape mode, so it prints the file lengthwise across the paper. Also, to make sure the text will fit, most applications let you preview your file before printing it.

4.5. Transferring files to or from research systems

The File Transfer Protocol (FTP) lets you **put** files that are on your computer onto another computer, or **get** files from another computer. It is very simple to use, and a few details are discussed next. This section focuses on transferring files between computing systems. Transferring files to and from research data storage systems is covered in "Accessing research data storage systems" on page 30.

4.5.1. From Windows

If you're on a Windows computer, you can transfer files to and from Big Red and Quarry using WinSCP or PuTTY.

4.5.1.1. WinSCP

1. Open WinSCP. In the `Host name` field, enter the host name of the server to which you are connecting (e.g., **`quarry.uits.indiana.edu`**).
2. In the `User name` field, enter your username for the host you specified.
3. In the `Password` field, type the password associated with the username you entered in the previous step.
4. Click `Login`.
5. When the software connects to your host, a window should appear with your local computer's directories on the left and the remote host's directories on the right. To move a file from your computer to the server, drag it from the left pane to the desired location in the right pane. To move a file from the server to your computer, drag it from the right pane to the desired location in the left pane.

4.5.1.2. PuTTY

1. To open the PSFTP part of the PuTTY application suite, from the **Start** menu, open **PuTTY**, and then **PSFTP**.
2. With the PSFTP login window open, type open followed by the name of the host you are connecting to (e.g., **`open quarry.uits.indiana.edu`**).
3. If the `Store key in cache?` prompt appears, type **`y`** .
4. At the login prompt, enter your username for the remote server.
5. At the password prompt, enter your password associated with the username you entered in the previous step. Once logged into the server, you will see the following command line prompt:

   `psftp>`
6. To display your current directory location, use the **`pwd`** command.

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

1. Use the **`cd`** command to navigate to the directory containing the file you wish to transfer. For example, **`cd documents`** will place you in the documents directory.
2. Use the **`lcd`** command to navigate to the directory on your computer file system to which you want to transfer the file (e.g., **`lcd documents`**).
3. To transfer the file, type **`get filename`**, replacing *`filename`* with the name of the file you wish to transfer.

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

1. Navigate to the directory on your computer file system containing the file you wish to transfer.
2. Navigate to the directory on the server file system to which you want to transfer the file.
3. To transfer the file, type **`put filename`**, replacing *`filename`* with the name of the file you wish to transfer.

Note: For a list of more commands, you can enter **`help`** or see *What is SFTP, and how do I use it to transfer files?* at:

http://kb.iu.edu/data/akqg.html

If you transfer a file to a Unix host, and the lines of text all end with ^M, it's most likely because Unix and Windows use different conventions for ending a line and beginning another. File transfer programs usually convert Windows text files to Unix automatically. You can also use the **dos2unix** utility to convert files:

```
dos2unix myfile mynewfile
```

4.5.2. From Mac OS

If you're working from a Mac OS workstation, we recommend using Cyberduck (which IU students, faculty, and staff can download for from IUware) to transfer files.

To use Cyberduck:

1. Double-click the Cyberduck icon on your computer. When Cyberduck opens, at the upper left, click **Open Connection**, or from the **File** menu, select **Open Connection**.
2. From the pull-down menu at the top of the sheet that appears, select **SFTP (SSH File Transfer Protocol)**.
3. In the Server: field, type the address of the remote host to which you wish to connect (e.g., ***quarry.uits.indiana.edu***).
4. In the `Login:` and `Password:` fields, type your Network ID username and passphrase. To save your password to the Keychain, check `Add to Keychain`.
5. If you want to log into a directory other than your home directory, click **More Options** at the bottom of the sheet. Type the directory name in the `Path:` field (e.g., ***www***).
6. Click `Connect` to open 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 `Allow` to accept the host's keys and continue connecting.)
7. A window will open displaying the list of files on the remote host. To upload files or folders, drag them from a Finder window into the Cyberduck window. To download files or folders, drag them from Cyberduck into the Finder.

4.5.3. From Unix

If your personal workstation is running a Unix operating system, including Mac OS X, you can use **scp** (secure copy) to transfer files. To transfer a copy of *mydata* from the present working directory on your workstation to your (*username*) home directory on Quarry, enter:

```
scp mydata username@quarry.uits.indiana.edu:
```

You will be prompted for your passphrase. After you enter it, you'll see an indication that the transfer was completed successfully.

To transfer a copy of **file.dat** from your home directory on Quarry to the current directory of your Unix workstation, enter:

```
scp username@quarry.uits.indiana.edu:file.dat  .
```

The dot at the end is important. It tells **scp** to put the file in the current directory, with the same name. Like **mv**, the **cp** and **scp** commands require source and target parameters. To make the same transfer, but change the name, enter:

```
scp username@quarry.uits.indiana.edu:file.dat ./newfilename
```

You can use scp to transfer files between supercomputers, too. For directions, see I*n Unix, how do I use the scp command to securely transfer files between two computers?* in the IU Knowledge Base:

https://kb.iu.edu/data/agye.html

4.6. Where should you keep your data?

Archival data storage is covered in "Archival and long term storage" on page 29.

You can store files on your home directory or in scratch space, or use a GPFS file system.

- **Home directory:** Your Big Red, Quarry and Research Database Complex (RDC) home directory disk space is allocated on the IBM N5500 NAS storage device. You have a 10 GB disk quota, which is shared with Big Red and the RDC if you have accounts on those systems.
- **Big Red local scratch:** Scratch disk space is available locally on each node in `/scratch` (67GB). Files in `/scratch` are automatically deleted once they are 14 days old.
- **Quarry local scratch:** Scratch disk space is available locally on each node in `/scratch` (19GB). Files in /scratch are automatically deleted once they are 14 days old.
- **RDC local scratch:** Scratch disk space is available locally on each node in `/tmp` (1GB) and an additional large scratch file system named `/scr` (10GB). Files in `/tmp` are automatically deleted once they are 24 hours old. Files in `/scr` are deleted when they are 30 days old. No disk quotas are enforced on `/tmp` or `/scr`.
- **Big Red, Quarry, RDC shared scratch:** Shared scratch space is accessible in one 346 TB GPFS file system, `/N/gpfs`. The path to your scratch space is `/N/gpfs/`*`username`*. On Big Red, files older than 180 days are purged automatically to free disk space; on Quarry and the RDC files older than 60 days are purged automatically.
- **Data Capacitor:** The Data Capacitor scratch directory is a temporary workspace. Scratch space is not allocated and its total capacity will fluctuate based on project space requirements. The Data Capacitor is mounted on Big Red and Quarry as `/N/dc/`... and behaves like any other disk device on that machine. If you have an account on Big Red or Quarry, you can access `/N/dc/scratch`. Files in scratch space may be purged after 14 days.

For more, see *At IU, how much disk space is available to me on the research systems?* at:

http://kb.iu.edu/data/avkm.html

Note: Scratch space does not count against your storage quota (the default at IU is 10GB).

Shared scratch space (`/N/gpfs/`) is intended as working space for running programs and temporary storage for program output. It's a high performance file system, mounted on all nodes. Local scratch disk space (`/scratch/`) is available only to the particular node with which it's associated – a distinction pertinent to multi-node jobs (also called parallel programs).

To use scratch space, create a directory. Enter:

```
mkdir /N/gpfs/username
```

You can upload data directly to that directory either by browsing to it with a graphical-based version of **scp** or **sftp**, or on a Unix system, enter:

```
scp ./filename username@quarry.uits.indiana.edu:/N/gpfs/username/
```

Files in scratch space are periodically purged to free up disk space, so it's important to maintain copies in other locations (see "Data storage" on page 29).

5. Running programs

To run a program, type its name:

```
$ date
Thu Aug 28 12:05:04 EDT 2006   (This is the computer's response)
```

5.1. Compiling Fortran, C, and C++ programs on Quarry

Note: If you use only commercial programs, such as SAS or MATLAB, you can skip to the next section ("Batch processing on Quarry").

To run a FORTRAN, C, or C++ program on any system, the source code must be compiled using that system's compilers, and then linked to that system's libraries.

On Quarry, you should use the Intel compilers named **ifort** and **icc**. To practice, you can copy a directory of example programs, and then compile one of them with the Intel compiler:

1. On Quarry, copy the directory. Enter:

   ```
   cp -r ~hpc/simple_quarry_jobs .
   ```

   Remember the dot on the end!

2. Move to the newly-copied directory:

   ```
   cd simple_quarry_jobs
   ```

3. Compile one of the sample programs (**sine.c**):

   ```
   icc sine.c -o sine_c
   ```

The third step compiles **sine.c** and outputs the executable (**sine_c**). To run it, enter:

```
./sine_c
```

The output will appear on your terminal.

```
Sin (0.000000) = 0.000000
Sin (0.392699) = 0.382683
Sin (0.785398) = 0.707107
Sin (1.178097) = 0.923880
Sin (1.570796) = 1.000000
```

5.2. Batch processing on Quarry

Quarry uses Torque, a version of the Portable Batch System (PBS), to manage jobs. Big Red uses a similar system, called LoadLeveler.

Important: If you have a program that runs more than 20 minutes, you must use the job management system to submit it for batch processing, so the system is used efficiently. You can use your own PBS submission script, or you can use **serialjob** (developed by Research Technologies staff).

5.2.1. Serialjob

You can run most single-processor commands as batch jobs using **serialjob** – serial meaning single-processor – which writes a PBS script for you and submits your job. If the command you want to run is in your current directory, simply insert **serialjob** before the command. For example, to run the program named **sine_c**, enter:

```
serialjob ./sine_c
```

If the program you want to run is not in the current directory, supply the full path to the command. Also, if it needs additional parameters, include them. For example, to run program **TheAnswer** with data file **data.dat**, enter:

```
serialjob /N/u/somedirectory/TheAnswer -d data.dat
```

Your program will be allowed to run for the default time allotted to jobs on the system. You can use **serialjob** on both Big Red and Quarry. For more, read the **serialjob** manual page. Enter:

```
man serialjob
```

5.2.2. *Submission scripts for the Portable Batch System*

If you don't use **serialjob** to generate your submission script (a file containing PBS directives), you must write your own.

You can use **cat** to examine a sample submission script for the example file we compiled in Section 4.1. The sample script is in the same directory of examples (`~/simple_quarry_jobs`). From that directory, enter:

```
cat submit_sine_c.sh
```

You should see:

```
#PBS -l nodes=1:ppn=1,walltime=5:00
#PBS -m ae
#PBS -q debug

${HOME}/simple_quarry_jobs/sine_c
```

This script consists of three PBS directives, each beginning with #PBS and followed by the "execution line:"

- The first line asks PBS for 1 node (`nodes=1`), and 1 processor per node (`ppn=1`), and 5 minutes of elapsed time to run it once it starts (`walltime=5:00`).
- The second line tells PBS to send an email notification when the job aborts or ends.
- The third line sends this job to the "debug" queue, which is intended for short jobs only. Without this line, the job would go to the "normal" queue.

The PBS directives come before any other commands – in this case the fourth line, which is the executable command. The **$** at the beginning isn't the shell prompt, it's a reference to the HOME environment variable – your home directory.

To submit a job to the PBS queue, use the **qsub** command:

```
qsub submit_sine_c.sh
```

This command will return the job's identification (jobid):

```
480005.qm2
```

To check the status of your submission, enter (where *username* is your username):

```
$ qstat -u username
```

A table will appear that looks like this:

```
                                                                  Req'd   Req'd     Elap
Job ID     Username   Queue     Jobname    SessID   NDS   TSK   Memory  Time   S  Time
---------- --------   -------   --------   ------   ---   ---   ------  -----  -  -----
480005.qm2  username   normal      --         --      1     1      --    0:05   Q  0:00
```

The status (S) column shows either `Q` (queued, waiting to run), `R` (running), or `C` (completed). When your job has completed, enter **`ls`**, and you will see your program's output file:

```
job_sine_c.o480005
```

For more information, see the manual pages, or search on "PBS" or "Torque" in the Knowledge Base.

5.3. Compiling Fortran, C, and C++ programs on Big Red

The Fortran, C, and C++ compilers on Big Red are named **xlf**, **xlc**, and **xlC**, respectively.

Except for very large programs, compiling is easily done interactively. Some elementary examples are located in the `~hpc/examples` directory. To view the contents, enter:

```
ls ~hpc/examples
```

You'll see the files:

```
helloWorld.C   helloWorld.f   llScript_C++
helloWorld.c   llScript_C     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, enter:

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

To practice with these programs, copy the `hpc/examples` directory to your home directory. Enter:

```
cp -r ~hpc/examples ~/
```

You can now compile your own executables. For example, enter:

```
cd ~/examples

xlf helloWorld.f -o helloWorld_F77
```

This tells Big Red to run **xlf** (the IBM Fortran compiler) with the input file **helloWorld.f**, and name the output file **helloWorld_F77**. If you don't specify an output file (**`-o`** *`filename`*), the system automatically uses **a.out**. When it's done, to run the output file, enter:

```
./helloWorld_F77
```

5.4. Batch processing on Big Red

Any job that uses more than 20 CPU-minutes must be submitted to LoadLeveler. The script **serialjob** will write your LoadLeveler submission script, as described previously in "Running programs" on page 21.

Sample LoadLeveler scripts (**llScript_F77**, **llScript_C**, and **llScript_C++**) are available in the directory of examples you copied in the previous section. If you didn't copy them before, copy them now. Enter:

```
cp -r ~hpc/examples ~/
```

To view one of the scripts, change to your new ~/examples directory, and use cat:

```
cd ~/examples
cat llScript_F77
```

You'll see:

```
#@ class = serial
#@ initialdir = ~/examples
#@ executable = helloWorld_F77
#@ output = helloWorld_F77.out
#@ error = helloWorld_F77.error
#@ queue
```

Note: The last line is "queue" – this is required.

It is common practice to create a newly named copy of such a file, and then modify it with an editor, such as **Nano**, when you have a new executable to run. For example:

```
cp llScript_F77 llScript_hostname
nano llScript_hostname
```

This copies `llScript_f77` to a new file named `llScript_hostname`, and then opens the file in Nano. In Nano, change the executable and output file names, as follows:

```
#@ executable = /bin/hostname
#@ output = hostname.out
#@ error = hostname.error
```

Note: If you are using a TeraGrid account, you must include this line (where `TG-your_account_number` is your TeraGrid account number):

```
#@ account_no = TG-your_account_number
```

To determine your TeraGrid account number, enter (where *TG_username* is your TeraGrid username):

```
tgusage -u TG_username
```

If more than one account number appears, use trial and error to determine which one works.

To submit `llScript_hostname` to LoadLeveler, enter:

```
llsubmit llScript_hostname
```

You'll see:

```
llsubmit: The job "s10c2b5.dim.823708" has been submitted.
```

To check the status of your submission, enter (where *username* is your username):

```
llq -u username
```

A table will appear that looks like this:

```
Id                       Owner     Submitted    ST  PRI Class     Running On
------------------------ --------- ------------ --  --- --------- ----------
s10c2b5.823708.0         username  12/1  12:30  I   50  LONG      s15c1b12
```

The status (`ST`) column shows either `I` (idle, waiting to run), `R` (running), or `C` (completed). When your job has completed, enter **ls**, and you will see your program's output file (`hostname.out`). Use **cat** to view the contents:

```
cat hostname.out
```

You'll see:

```
s15c1b9
```

This indicates that the job, submitted from the node named **s10c2b5**, ran on node **s15c1b9**.

Some useful LoadLeveler commands include:

`llclass`	Describes the defined batch classes (queues)
`llq`	Queries the status of running and queued jobs
`llstatus`	Displays the status of LoadLeveler machines
`llsubmit`	Submits your script to LoadLeveler
`llcancel`	Cancels a running or queued job
`showq`	Shows information about job scheduling
`showres`	Displays job reservations

For more about LoadLeveler, including several example scripts, search on "loadleveler" in the Knowledge Base. For online documentation, see:

https://kb.iu.edu/data/azvs.html

If your source code consists of more than one file, or requires compiler options to improve runtime, or if your research might benefit from improving the performance of your program (e.g., by parallelizing your source code), please contact the High Performance Applications team.

5.5. Softenv

Big Red and Quarry both use the **Softenv** environment management system to simplify environment configuration. When you log in for the first time, a `.soft` file is created in your home directory, which configures your default environment, including compilers.

You may add additional packages if you wish. To temporarily add a keyword to your environment, enter:

```
soft add +keyword
```

To get a list of the possibilities for *keyword*, enter:

```
softenv
```

To restore your environment to the default settings in your `.soft` file, enter:

```
resoft
```

To permanently change your environment, edit your `~/.soft` file (using an editor such as Nano). For example, if you wish to run SAS and MATLAB jobs, add keywords as follows:

```
# This is your SoftEnv configuration run control file.
#   It is used to tell SoftEnv how to customize your environment by
#   setting up variables such as PATH and MANPATH.  To learn more
#   about this file, do a "man softenv".
#
@quarry
+Matlab
+sas
```

To make the changes take effect, enter:

```
resoft
```

Or, log out, and then log back in.

5.6. Commercial programs on Quarry

The popular commercial programs SAS and MATLAB are available on Quarry. As noted in the previous section, Quarry uses Softenv to manage the software environment. To run SAS or MATLAB, you must add the appropriate keyword to your `.soft` file. Short jobs can run interactively, but large jobs (those requiring more than 20 minutes of CPU time) require you to use a job management system, as introduced above in "Running programs."

5.6.1. SAS

Sample SAS program and data files are available on Quarry. To copy them to your home directory, enter:

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

A SAS program normally contains a data file and a program file. For example, the following program file 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):

```
DATA sample;
INFILE '~/scripts/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 your root directory. If the data file is stored in another directory (e.g., `ingen`), you must specify the full pathname:

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

The data file is stored 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
```

5.6.1.1. Running SAS Jobs

Once you have the program and data files, you can choose from three methods to execute your job: interactive, background, and batch.

5.6.1.2. Interactive and background jobs

If your SAS program can complete in less than 20 minutes, you can run it interactively. To run the above program interactively, enter:

```
sas sample.sas
```

Once the program has executed without errors, two additional files will be written to your directory (**sample.log** and **sample.lst**). The .log file contains log information for your program, and the .lst file contains

the output listing. If a .lst file is not created, examine the .log file for error messages. If there are error messages, edit the program file accordingly, and then rerun the program.

Note: If you're running an SAS program interactively, you may not be allowed to continue until the job is complete.

If you want to continue computing while your program is running, you can execute it as a *background process*. Your program will run while the shell returns you to the system prompt. To run your SAS program as a background process, enter the same command as before, but add a trailing ampersand:

```
sas sample.sas &
```

To list programs running in the background, use the **jobs** command. When your program has completed successfully, the .log and .lst files will be written to the directory.

5.6.1.3. Batch jobs

If your SAS program will use more than 20 minutes of processor time, or if you need large amounts of memory, you must use the batch system. You can do this easily using the **serialjob** command:

```
serialjob sas sample.sas
```

The **serialjob** script creates a PBS or LoadLeveler script for you and submits it. You should run **serialjob** from the same directory that acts as the current directory when the job runs, and make sure your SAS code and data files are in that directory, as well.

Instead of using **serialjob**, you can write your own PBS (on Quarry) or LoadLeveler (on Big Red) script, as described in earlier sections.

5.6.2. MATLAB

You can run a MATLAB program on Quarry using three methods: interactively, as a background process, or by submitting the program to PBS as a batch process. You must run any program that requires more than 20 CPU-minutes as a batch job.

5.6.2.1. Interactive and background processes

To start MATLAB, enter:

```
matlab
```

Your terminal will display the MATLAB "splash screen" and prompt:

```
< M A T L A B ® >
Copyright 1984-2009 The MathWorks,Inc.
Version 7.8.0.347 (R2009a) 64-bit (glnxa64)
Febr
uary 12, 2009

>>
```

Enter MATLAB commands at the prompt (>>). For example, to produce a four-by-four Hilbert

matrix, enter:

```
hilb(4)
```

The program will respond 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
```

To quit MATLAB and return to the system prompt, enter:

```
quit
```

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

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

You can also run MATLAB as a background process. To do so, you must first create a text file (using a text editor, such as Nano) containing the commands you want MATLAB to run (e.g, `matlabinput`). To feed these commands to MATLAB, and display MATLAB's output on the terminal screen, enter:

```
matlab < matlabinput
```

To run MATLAB in the background, and have the output written to a file (e.g., `matlaboutput`) and any operation system errors written to another file (e.g., `matlaberror`), enter:

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

Note: The ampersand at the end of the line tells the computer to run MATLAB in the background.

After the program finishes, look for the `matlaboutput` and `matlaberror` files (the error file should be empty). You should also see a `results.mat` file, which records the results in a format MATLAB can load later.

5.6.2.2. Submit a MATLAB batch job

To submit a MATLAB batch job on the Quarry cluster at Indiana University, create an m-file with the commands for MATLAB to run (e.g., `matlab_input.m`). Then write a script file (e.g., `matlab_job`) for TORQUE (also called PBS), the resource manager on Quarry, as follows:

```
#!/bin/bash
#PBS -l nodes=1:ppn=1,walltime=30:00
#PBS -M username@indiana.edu
#PBS -m abe
#PBS -N JobName
#PBS -o matlab_output
#PBS -e matlab_error
matlab -r matlab_input
```

Note: If you are not working in the default directory (i.e., `/N/u/username/Quarry`), you need to change the directory in the script (e.g., **`cd /N/u/username/Quarry/new`**) and matlab_input (e.g., '**`~/new/`**') files.

To submit the script file to TORQUE, from the command prompt enter:

```
qsub matlab_job
```

The default queue is the LONG queue. For more on queues and how to specify them, see "At IU, how do I use TORQUE/PBS on Quarry?" at:

https://kb.iu.edu/data/avmy.html

5.6.2.3. Check job status

To check the status of your job, use either the `qstat` or `showq` command. After the job is executed, you will receive an email confirmation.

For more information about using SAS, MATLAB, or any other statistical and mathematical computing software, contact the Stat/Math Center at 812-855-4724 or 317-278-4740 (statmath@indiana.edu), or visit the Stat/Math web site:

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

6. Data storage

6.1. Archival and long term storage

IU's supercomputers provide multiple file systems. Home directories (where you keep source code and small files) and scratch space (where you store data temporarily) were covered in "Where should you keep your data?" on page 20. This section is devoted to long-term storage of large datasets.

The Research Storage group in the UITS Research Technologies division administers and supports the Research File System (RFS) and the Scholarly Data Archive (SDA):

- The **Research File System** is a spinning disk system with currently 30TB of total capacity. RFS can be mounted on the desktop or accessed over the web or SFTP protocol. RFS supports active editing of files and documents unlike the SDA. Files are backed up on a nightly basis from RFS. RFS has robust support for project work as well. Quotas start at 10GB for personal space and 50GB for project space. The Scholarly Data Archive (SDA) provides extensive capacity (4.2 petabytes) for storing and accessing huge amounts of research data. It uses the High Performance Storage System (HPSS), a hierarchical storage management software package. The system is located at both Indiana University Bloomington and IUPUI, providing automatic off-site copies of data for disaster recovery. Individual users can have up to 5 TB of space.
- The **Scholarly Data Archive** is primarily a tape based archive storage system. Currently SDA has a raw tape capacity of 5.7PB. It does have over 200TB of spinning disk that operates as a cache for files moving to or from tape. SDA is geographically distributed between the IUPUI and IUB campuses. By default, a file stored in SDA will have a copy on each campus. This protects from minor problems like tape failures to major ones like a site disaster. Each campus has an automated tape library capable of holding over 5,000 tapes and 24 high speed tape drives. SDA has an aggregate transfer rate of over 2GB per second. It is capable of storing files from about 1MB to over 5TB is size.

6.1.1. Requesting accounts

To request an RFS or SDA account use the online Account Management Service (AMS):

https://itaccounts.iu.edu/

To request a quota increase, email store-admin@iu.edu.

Note: To establish a group, lab, or departmental account, you will need to request an appropriate Network ID – also done via the AMS. Once you have the group Network ID, use it to log into the AMS, and then request a storage account.

Undergraduates who want accounts on RFS or SDA must have faculty sponsorship for specific research projects. Sponsors should email valid@iu.edu with reasons for undergraduate account requests.

6.1.2. Accessing research data storage systems

A number of methods are available for transferring data to and from the Scholarly Data Archive (SDA) at Indiana University. These include Kerberos-enabled FTP, parallel FTP (pftp_client), Hierarchical Storage Interface (HSI), secure FTP (SFTP), secure copy (scp), SMB/CIFS/Windows file sharing (SMB), and, with a web browser, https. See:

- At IU, how do I use parallel FTP to transfer data to or from the SDA?
  https://kb.iu.edu/data/aukx.html
- At IU, how do I use HSI to access my SDA account?
  https://kb.iu.edu/data/avdb.html
- At IU, how do I map or mount my SDA account to my workstation?
  https://kb.iu.edu/data/auxm.html
- At IU, how do I use the Scholarly Data Archive web interface?
  https://kb.iu.edu/data/auxl.html
- At IU, how do I use SFTP or SCP to access my SDA account?
  https://kb.iu.edu/data/avax.html

For more information, see the SDA Starter Kit at:

https://pti.iu.edu/storage/mdss-starter-kit

6.1.2.1. Mounting RFS on your desktop system

Once your RFS account is created, you'll be able to access your files and folders as if they were stored on your workstation's hard drive. For complete instructions see "At IU, how do I map or mount my RFS account to my workstation?" at:

https://kb.iu.edu/data/arxp.html

6.1.2.2. Hierarchical Storage Interface

Hierarchical Storage Interface (HSI) and its companion program, HTAR, can simplify aggregation of many files into one large file, which is the preferred method of storage in HPSS. HSI commands should be familiar to Unix and FTP users.

On the Research Database Complex, HSI is installed in `/usr/local/bin`, so it's in the default path. On Big Red and Quarry, you may initially need to add **`+hpss`** to the end of your `.soft` file, and then use the **`resoft`** command to make it available. To use HSI on your personal workstation, download a copy from the HSI/HTAR version 3.5.3 directory, https://rfs.iu.edu/clients/clients/3.5.3/ (IU Network ID credentials required).

A sample session follows (`$` is the Unix shell prompt and `?` is the HSI prompt):

```
$ hsi
Principal: jdoe
[jdoe]Password:
Username: jdoe  UID: 11021  CC: 11021 Copies: 1 [hsi.3.3.3 Fri
Jan 12 13:36:06 EST 2007]

? ls
/hpss/j/d/jdoe/:
NPB-ppcc.tar   foobar/        movies/

? du -k
861309  /hpss/j/d/jdoe/
-----------------------
861309  total 1024-byte blocks, 6 Files (881,979,719 bytes)

? put myfile1.dat
put myfile1.dat : /hpss/j/d/jdoe/myfile.dat ( 10485760 bytes,
```

```
12283.4 KBS (cos=3))

? cd movies
? get myfile2.mov
Scheduler: retrieving file(s)
get myfile2.mov : /hpss/j/d/jdoe/movies/myfile2.dat
(2005/09/29 08:49:03 10485760 bytes, 16842.8 KBS )

? exit
```

For more, see the HPSS Hierarchical Storage Interface (HSI) Online Documentation at:

https://pti.iu.edu/storage/hpss-hsi-online-documentation

6.2. High speed temporary storage

Created in part to meet the demands of modern electronic sensors, IU's Data Capacitor is a high-speed file system that has achieved world-record simultaneous read/write performance (approaching theoretical limits).

The Data Capacitor is not intended for permanent storage of data, and is not backed up. You can archive Data Capacitor data on the Scholarly Data Archive (SDA) using, for example, HSI. It is your responsibility to arrange for long-term permanent storage of any data on the system as needed.

The Data Capacitor is mounted on Big Red and Quarry at the mount points `/N/dc/` and `/N/dcwan/`, where it behaves like any other disk drive. All Big Red, Quarry, and TeraGrid users have scratch space directories available at `/N/dc/scratch/`*`username`* (replace *`username`* with your username). Space is not allocated, so the amount of storage varies, but terabytes are available.

You may also request long-term project space. The default allocation is 10TB, but projects may be awarded more than 10TB if justifiably needed. When you have allocated project space, you'll access it at `/N/dc/projects/`*`projectname`* (replace *`projectname`* with your project's name).

Users at other institutions (including IU researchers with accounts on remote systems) can request Data Capacitor storage space, which can be mounted at the remote institution, as well as on Big Red and Quarry. Access to the WAN space is available at `/N/dcwan/`.

7. Visualization facilities

In addition to understanding your analysis and how to best represent your data visually, you need to consider how visualization tasks factor into your scientific workflow. In the most common scenario, you 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, but all computation and visual display occurs on your workstation.

In some instances, your visualization and analysis work may require network access to large amounts of data from the mass store system, parallel computation from a supercomputer, or access to remote hardware or instruments. Working in conjunction with other UITS teams, Research Technologies visualization staff can help design a system that maximizes your efficiency.

In other instances, your visualization work may benefit from special-purpose hardware, and computing systems with more memory or processing power than your local system. In such cases, you may use the resources of the Advanced Visualization Laboratory (AVL), located at IUPUI in the ICTC Building, and at IUB in Lindley Hall.

For more information see:

http://avl.iu.edu/

7.1. Supported visualization platforms

The AVL's primary supported visualization platforms are Windows and Linux, but they also can provide limited support for Mac OS X and IRIX. Whenever possible, AVL staff try to identify and adopt tools that are platform independent; however, special-purpose displays, hardware cards, and software packages may impose restrictions.

7.2. Visualization services

Software consulting. The most effective way to get started with visualization (or to enhance your current work with visualization) may be to arrange a consultation with AVL staff (email vishelp@indiana.edu). Staff may be able to help you derive more functionality from the applications you are currently using, or help you identify tools best suited to your needs. They can also assist with data translational tasks, and application optimization through custom scripts, macros, or plug-ins.

Custom software development. If existing software tools are not suited to your visualization task, the AVL may be able to help you develop a custom application that precisely fits your needs. While such development may take several months or more, the result can be extremely valuable. The AVL has developed custom applications for volume rendering, molecular, phylogenetic and pedigree tree, and information visualizations.

Hardware consulting. While central facilities are satisfactory for occasional use, one-time demonstrations, or technology "test drives," visualization technologies have the greatest impact when they are conveniently and directly accessible to their users. AVL staff can help you derive specifications for systems that best fit your needs, including desktop workstations, specialized hardware, and displays. The AVL maintains good working relationships with several vendors, should outside help be necessary.

7.3. Visualization software

The AVL's key goal is identifying, exploring, and providing access to the latest and most advanced visualization software. The AVL conducts extensive investigations into a wide range of frameworks, APIs, and software tools that may prove useful for the visualization needs of the university community. Preference is given to open source or free tools, but there are situations in which proprietary tools are most appropriate. The AVL hosts its own license server, and can provide a limited number of licenses for key visualization software packages, including (but not limited to) VirTools, CAVELib, Maya, 3D Studio Max, Rapidform, and ZEdit Pro. Open source, free, and/or locally developed codes include (but are not limited to) VTK, ParaView, X3D, 3DIVE, PathView, XMView, Effect, and OpenGL. The AVL's list of recommended software is vast and ever changing. You're encouraged to schedule a consultation with an AVL team member to ensure you are using the most effective software tool for your needs.

7.4. Visualization facilities

While you can carry out many visualization tasks on a desktop workstation, some tasks might require highly specialized hardware, or have computation or memory requirements that exceed those of a desktop system. For these situations, the AVL maintains central facilities at IUB and IUPUI, which provide range of advanced technologies, including (but not limited to) virtual reality and high resolution displays, advanced interface/input/output devices, the latest GPUs and rendering hardware, and a selection of teleconferencing and tele-collaboration systems.

For a specific listing and more detailed information, see:

http://pti.iu.edu/avl/facilities

A final note...

The information in a document like this can become outdated quickly, and must be revised frequently. Your insights are invaluable to improving the information we deliver in the future. **Was anything missing? Was anything unclear? Was there more than anyone needs to know?**

We take your advice seriously. Please send your comments to researchtechnologies@iu.edu.

Acknowledgments

We gratefully acknowledge the support of the National Science Foundation; the Lilly Endowment, Inc.; the National Institutes of Health; IBM, Inc.; StorageTek; and Sun Microsystems, as well as support from the state of Indiana and the university. Without the generosity of these organizations, we would not be able to offer the systems and services outlined in this document. The Lilly Endowment's support for the Pervasive Technology Labs, the Indiana Genomics Initiative, and METACyt has been particularly important in advancing IU's strategic goals in advanced information technology and life sciences.

Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect the views of those organizations.

Contributing authors

Andrew Arenson
Michael Boyles
Robert Cruise
Arvind Gopu
David Hart
Peg Lindenlaub
Mary Papakhian
John Samuel
Kurt Seiffert
Anurag Shankar
Craig Stewart
Eric Wernert

Editorial assistance
Paul Brown
Malinda Lingwall
Paul Messing
Sean Pendergast

Appendix 1. Contact information

For **general purpose computing support** (Windows, Mac OS X, Linux, Unix), contact the Support Center (24/7):

Email: ithelp@iu.edu

Phone: IUB – (812) 855-6789

IUPUI – (317) 274-4357

For online help, search the IU Knowledge Base:

https://kb.iu.edu/

The UITS Research Technologies division is a part of the Pervasive Technology Institute:

http://pti.iu.edu

Applications:

http://pti.iu.edu/rta

- High Performance Applications hpa-admin@iu.edu
- Open Science Grid Operations goc@opensciencegrid.org
- Science Gateways Group mpierce@cs.indiana.edu

Life sciences:

http://pti.iu.edu/rtl

- Advanced IT Core rtls@iu.edu
- Biomedical Applications rtls@iu.edu
- Bioinformatics and Computational Biolgy rtls@iu.edu

Systems:

http://pti.iu.edu/rts

- Core Services rtadmin@rtinfo.indiana.edu
- Data Capacitor dc-team-l@indiana.edu
- High Performance Systems hps-admin@iu.edu
- Research Storage store-admin@iu.edu

Digital library development:

Email the Digital Library Program at diglib@indiana.edu, or see:

http://pti.iu.edu/dlp/

Statistical and mathematical computing:

Call the Stat/Math Center at 812-855-4724, email statmath@indiana.edu, or see:

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

Visualization:

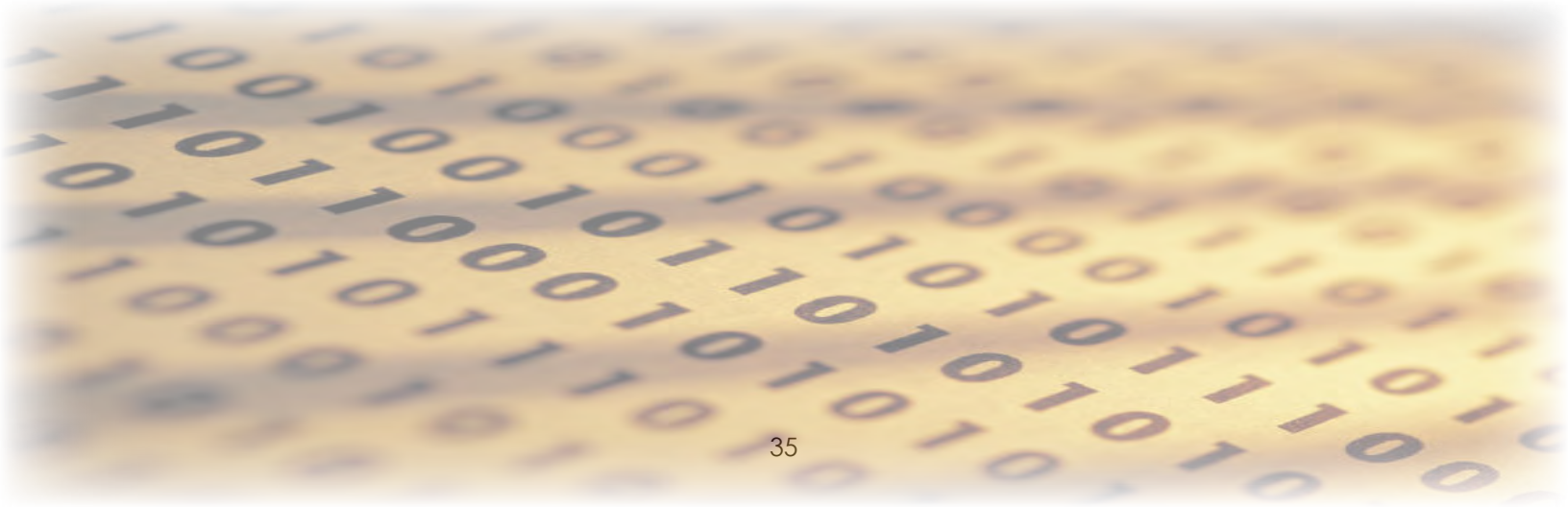
Email the Advanced Visualization Lab at avlstaff-l@indiana.edu, or see:

http://pti.iu.edu/avl

Appendix 2. Terminology: Bits, bytes, and FLOPS

Over time, the computing community has developed its own private dialect. You don't need to memorize these terms to understand this document, but it's helpful to know how data sizes and performance measurements compare:

- A **bit** is the fundamental (and smallest) unit of data representation within a computer. A bit is always either a 0 or a 1.
- A **byte** is the smallest unit of data representation usually used by a computer. A byte consists of 8 bits, and represents roughly one alphabetical character (such as A or B), one digit (such as 2 or 7), or one punctuation mark (such as a comma). Eight bits provides 28 (256) characters in the standard English computer alphabet.
- A **megabyte (MB)** is one million bytes (or one million characters). There are about 700 megabytes of data on a typical CD.
- A **gigabyte (GB)** is a billion (109) characters.
- A **terabyte (TB)** is a trillion (1012) characters. All the printed text in the IUPUI library is approximately 1TB.
- A **petabyte (PB)** is a quadrillion (1015) characters. The printed text in all US academic libraries is approximately 2PB.
- **FLOPS** stands for Floating Point Operations Per Second. A floating-point operation is multiplying two decimal numbers together (such as 2.78 x 3.14).
- **Gigaflops** represents one billion (109) FLOPS. If a computer performed one floating-point operation per clock tick, then its FLOPS count would always be the same as the clock rate of its processor chip. In practice, however, this isn't so. A computer's theoretical peak FLOPS can be faster if it has processors that perform more than one mathematical operation per clock tick. In any case, a computer never achieves its theoretical peak FLOPS count.
- **Teraflops** represents one trillion (1012) FLOPS.



INDEX