ICESAT/GLAS
Science Computing Facility

Linux Installation Guide

Kristine Barbieri

Steven McLaughlin

Steve Fiegles

Version 200907.0
July 2009

NASA/Goddard Space Flight Center
Cryospheric Sciences Branch/Code 614.1
Greenbelt, MD 20771

Section 1: Introduction

This document outlines the system requirements and setup procedures for creating a GLAS Remote Science Computing Facility (rSCF) on a Linux system. The steps in each section must be completed before progressing to the next section.

Section 2: Table of Contents

SECTION 1: INTRODUCTION 2

SECTION 2: TABLE OF CONTENTS 3

SECTION 3: RSCF SYSTEM CONFIGURATION 4

SECTION 4: RSCF SYSTEM SETUP BEFORE SOFTWARE INSTALLATION 6

SECTION 5: INITIAL SOFTWARE DELIVERY FROM ICESAT6 7

SECTION 6: MSCF SYSTEM SETUP 8

SECTION 7: RSCF SYSTEM SETUP AFTER SOFTWARE INSTALLATION 10

SECTION 8: SCF SOFTWARE DELIVERY PROCEDURE 11

SECTION 9: PROBLEMS WITH PORTING THE SCF SOFTWARE 12

APPENDIX A: CREATING SSH KEYS FOR NON-INTERACTIVE DATA EXCHANGE 16

APPENDIX B: INSTALLING MYSQLTCL FOR SITES THAT ACCESS A MYSQL DATABASE 17

APPENDIX C: RECOMMENDATIONS ON SELECTING A LINUX OPERATING SYSTEM 18

Section 3: RSCF System Configuration

Hardware/Software requirements:

- PC running Linux
- RedHat Enterprise Linux 4 WS ($1000-$2000), CentOS (free), or Ubuntu (free)
  (Refer to Appendix C for recommendations on selecting a Linux OS)
  - 32 or 64 bit OK
  - 500GB disk space - 1TB RAID5 preferred
  - 1GB memory
- IDL V6.4 ($2400 - less for gov, edu)
- G95 Fortran 95 compiler (free from http://www.g95.org/)
  - 64-bit compiler with 32-bit default integer
- GCC compiler (included with OS package, although may need to be optionally installed).
- Total Cost ~$5k
- Please ask John DiMarzio if you need configuration assistance

The System Administrator needs to setup the following on the Linux system:

- An "scf" account assigned to groups "external" and "users"
  - Contact the SCF manager for uid and gid information
  - Give the SCF manager the temporary password and she will change it immediately
- Home directory /SCF owned by scf:external
- Entry in /etc/passwd for scf account must have group id=401 and home directory as /SCF:
  - i.e. scf:x:1005:401:NASA ICESAT GLAS:/SCF:/usr/bin/ksh
- Entry in /etc/hosts for icesat0
  - Contact the SCF manager for IP address
- Capability to send mail from the scf account
- Nedit (RedHat binary software distribution or http://www.nedit.org/)
- System path must include /opt/bin
- IDL: Executable link to /usr/local/bin/idl
- Tcl: Executable link to /usr/bin/tclsh
- Perl: Link /usr/bin/perl to /opt/perl/bin/perl
- Gunzip: Link /bin/gunzip to /usr/contrib/bin/gunzip
- Ssh: Version 2 running as /usr/bin/ssh
- Link /SCF/bin/ops/run_visualizer.ksh to /opt/bin/glas_visualizer
- Link /SCF/bin/ops/run_data_request.ksh to /opt/bin/glas_data_request
- Crontabs running under root and scf
- Root crontab running /SCF/bin/clean_scf_tmp.ksh:

  0 1 * * 3,6 /SCF/bin/clean_scf_tmp.ksh >> /SCF/tmp/rootcronlog #Clean up old directories under /SCF/tmp

- Partitions for /SCF and /SCF/product_sets (data)

Once the Linux system has been configured as above, contact the SCF manager with the following information:

- Host name and IP address of the new Linux system
- Temporary password for the scf account and she will change it immediately
- Name and email address of the person with root access to the Linux system

Note to g95 users:

During runtime, you may get a message like the following:

> Remaining memory: 5280 bytes allocated at line 169 of /SCF/src/common/200611.0/ANC70_mod.f90

To suppress these warnings, you need to put the following line in the .kshrc file:

```
export G95_MEM_SEGMENTS=0
```

During runtime, you may get a message like the following:

> /SCF/bin/ops/read_product_file: error while loading shared libraries: libexec.so: cannot open shared object file: No such file or directory

To suppress these warnings, you need to put the following line in the .kshrc file:

```
export LD_LIBRARY_PATH=/SCF/lib/ops
```

Section 4: RSCF System Setup before Software Installation

The SCF Manager will set up the following on the Linux system before the initial software is installed:

1) If the scf account was setup with a temporary password, change it to the mSCF scf password by typing: passwd.

2) Check the default shell by typing: echo $SHELL

   If the default shell is not korn shell (i.e. /usr/bin/ksh), then change it by following these steps:

   - Type: chsh -l

     This gives you a list of the available shells and the appropriate pathname for each. The path to the korn shell should be listed: /bin/ksh

   - Type: chsh

     User is asked for the password, and then for the new shell path--which you get from the list above.

     NOTE: This seems to be somewhat different than the way things work in UNIX, which recognizes the chsh command but not the -l option.

3) Scp the following files from icesat0:/SCF to the /SCF directory on the new Linux system:

   - .profile_linux (needs to be renamed to .profile on the new Linux system)
   - .kshrc_linux (needs to be renamed to .kshrc on the new Linux system)
   - .idl_startup

   Log out and log back in to test. Should see host name and current directory as the prompt.

   i.e. falcon:/SCF

4) Create the ssh keys between the new Linux system and icesat0 (Refer to Appendix A for instruction on how to do this)

Section 5: Initial Software Delivery from icesat6

The software that needs to be delivered comes in two parts. The first delivery, as described in this section, consists of the initial Linux software that only needs to be installed once. Since icesat6 is the mSCF Linux test bed machine, the initial Linux software comes from this system. The subsequent software delivery will be described in Section 8 and comes from the mSCF system computer, icesat0, which is HPUX. The software in this delivery is updated every time there is a new SCF software release (usually corresponding to GLAS Data Product releases).

If the tar file from icesat6 doesn't need to be updated:

1) The tar file can be found in icesat0:/SCF/dist/tar_files/SAT6

If the tar file needs to be updated, the SCF Manager will perform the following on icesat6:

2) Create a tar file on icesat6:
   - Go to the /SCF/tar_files directory on icesat6.
   - Edit the new make_tarfile script to reflect the new tar file name and the directories and files that need to be tarred.
   - To run the make_tarfile script to create the tar file, enter the command:

     make_tarfile

   - Use the scp command to copy the tar file to icesat0:/SCF/dist/tar_files/SAT6. For example:

     scp scf_sat6_200708.tar icesat0:/SCF/dist/tar_files/SAT6

3) From icesat0:/SCF/dist/tar_files/SAT6, use the scp command to copy the tar file to the /SCF directory on the new Linux system. For example:

   scp scf_sat6_200708.tar lidar-rscf.cnr.colostate.edu:/SCF

The SCF Manager will set up the following on the Linux system:

4) In the /SCF directory on the new Linux system, untar the tar file. For example:

   tar xvf scf_sat6_200708.tar

5) Ensure that the /SCF/tmp directory is write-accessible by all by typing:

   chmod 777 /SCF/tmp

Section 6: MSCF System Setup

The SCF Manager will perform the following on the main SCF (icesat0):

Note: If the institution is not new and already has an abbreviation, and is just replacing an old HP machine with a new linux system, then you only need to do the items in orange.

1) Determine an abbreviation for the new system based on the institution or host name (ABBR). By convention, use capitals such as "LRSCF" or "SCFWEB" when referring to directories and the same letters but as lower case when it appears in a related file name.

2) Add the following directories:
   - /SCF/dist/ABBR
   - /SCF/ingest/ABBR

3) Modify the Perl transfer scripts for the new system:
   - In /SCF/src/perl:
     - Create a new directory ABBR
     - Copy everything under SAT6 to ABBR (include all sub-directories)
     - Add the new ABBR system to the following scripts:
       - create_new_perl_version.ksh
       - scp_perl_sites.ksh
   - In /SCF/src/perl/ABBR/ops:
     - Edit run_scripts.ksh as necessary to modify the host computer identifications and as per the accompanying README file.
     - Verify that the .pl and .pm files are linked to the files in the most recent dated-version subdirectory.
   - In /SCF/src/perl/ops:
     - Copy run_scripts_sat6.ksh to run_scripts_ABBR.ksh and edit as necessary to modify the host computer identifications
     - Add the new system to the following scripts:
       - run_cleanup.ksh
       - get_subdir.pm – copy to /SCF/src/perl/ABBR/*latest_version*
     - Copy any new or modified files to /SCF/src/perl/working as well

4) From the new directory, copy all files and subdirectories to the new Linux system. For example from /SCF/src/perl/ABBR type the command:

   scp -r * lidar-rscf.cnr.colostate.edu:/SCF/src/perl

   Verify that the files in the ops subdirectory are linked to the files in the most recent dated-version subdirectory after the copy. If they are not, then edit the version in /SCF/src/perl/perl_link_to_ops.ksh and run.

5) Add the new system to the following scripts:
   - /SCF/utils/scf_site_hosts.ksh (add to list and to LINUX_SITES alias)
   - check_ps_linux_rscf.tcl in /SCF/bin/ops and working (add to environmental variable list and to site_list and site_host_list).

- check_ps_rscf.tcl in /SCF/bin/ops and working (remove from environmental variable list and from site_list and site_host_list).
- run_check_dist.ksh in /SCF/bin/ops and working
- test_check_dist.ksh in /SCF/bin/ops and working
- create_maps.pl in /SCF/src/web/ops and working
- get_area_dates.pm in /SCF/src/web/ops and working
- subscription.pl in /SCF/src/web/ops and working

6) Add a new database table to ops and working MySQL databases: ABBR_PRODUCT_ID

7) Add ABBR to the data request GUI software in setdefaults.pro in /SCF/IDL/data_selection/ops and working

8) Add a new entry to the crontab:

   * * * * * /SCF/src/perl/ops/run_scripts_ABBR.ksh

9) Add the new site to the subscription statistics page on the SCF website:
   - In glas-scfweb.gsfc.nasa.gov:/var/www/html/Statistics/src:
     - Add the new site ABBR to daily_stats_2007.pl as necessary (remember to also Increment "i" counter: search $i=0, it's the fourth one)
     - Increment "i" counter in create_summary_chart.pm
     - Add site ABBR to create_summary_page.pm. If more charts need to be created, increment "i" (columns) or "j" (rows)
     - If need to change "Yearly Distribution Summary" window size, change the height and width values in function "fn1" in create_calendar_intro_2007.pm. You can play with these values beforehand in /var/www/html/Statistics/stat_2007/html/2007_calendar.html (this file is created by create_calendar_intro_2007.pm and gets overwritten).
     - Type ./run_daily_stats_perl.ksh to run script
   - Check on SCF website under "Statistics"-"2007 Calendar". Click on a date to see if the new site is included in the plots. Also check the "Yearly Distribution Summary".

Section 7: RSCF System Setup after Software Installation

The SCF Manager will set up the following on the Linux system after the initial software is installed:

1) Create the following SCF crontabs, which are a list of commands that execute automatically at pre-selected times. This is most easily accomplished using the "-e" option of the "crontab" command and a copy from the icesat6 crontab listing ("-l" option). Remove the "#" signs to uncomment commands.

   ```
   * * * * *  /SCF/src/perl/ops/run_scripts.ksh

   00 * * * * /SCF/bin/ops/run_check_dist_rscf.ksh >> /SCF/tmp/cron_dist_errors.txt 2>&1
   ```

2) Deliver the newest SCF software (See "SCF Software Delivery Procedure" in Section 8)

3) Add new site info to site_hosts.doc

4) Add new system and contact info to rscf_contacts.doc

Section 8: SCF Software Delivery Procedure

The following steps are to be performed by the SCF Manager on the Linux system every time there is a new SCF software delivery:

1) A new tar file has to be created in icesat0:/SCF/dist/tar_files/LATEST_VERSION to include changes just made. To do this:
   - Create a LATEST_VERSION subdirectory under icesat0:/SCF/dist/tar_files/
   - Copy make_tarfile from the latest previous subdirectory and edit as necessary
   - Type make_tarfile to create the tar file

2) Deliver the tar file to the /SCF directory on the Linux system:

   i.e. scp /SCF/dist/tar_files/200703.0/scf_200703.tar hudson:/SCF

3) In the SCF directory, untar the tar file by typing: tar xvf <tarfile>

4) In /SCF/src/readGLAS/<latest_libvers> remove the *.a file (libgsas.a).

5) Copy /SCF/cc_util/* to /SCF/src/readGLAS/<latest_libvers>

6) Modify the library and version in make_vers.incl in the following directories:
   - /SCF/Make
   - /SCF/src/readGLAS/Make

Note: An easy way to do this is to edit the version of make_vers.incl in icesat0:/SCF/utils/linux_files, then copy it to the linux sites with /SCF/utils/scp_file_linux_sites.ksh.

9) In /SCF/Make type “make” to create Fortran executables.

Note: If the make errors out with a time_lib error, scp icesat0:/SCF/utils/linux_files/dateinterface.c to the /SCF/libsrc//<latest_libvers>/common_libs/time_lib directory on all the linux sites using /SCF/utils/scp_file_linux_sites.ksh.

10) In /SCF/lib/ops link the latest libraries by typing: ln -s ../V5.3/* . (if V5.3 is latest library).

11) In /SCF/src/modules/ops link the latest modules by typing: ln -s ../V5.3/* . (if V5.3 is latest library).

12) Test the delivery by running the following:
   - /SCF/lib/ops/read_product_file
   - /SCF/src/readGLAS/<latest_libvers>/read_glas_file and read_gla01_wf
   - /SCF/product_sets/SRTM_test/high_dem_ret
   - glas_data_request - submit the request to ensure that the data delivery system works
   - glas_visualizer

Section 9: Problems with Porting the SCF Software

The SCF software was originally created on a HP UX system and several issues were discovered while porting the software to run in the Linux environment. Some of the problems we had using the Absoft f90 compiler were fixed when we switched to the g95 compiler.

C Code

- **Makefiles**

When using the Absoft compiler, the c preprocessor required capital file names for the Fortran code suffixes of the form ".F90", whereas the HP Fortran 90 code exists with suffix ".f90". Without the capitalized form, the preprocessor was not being evoked and all diagnostics in the HP code were falling out as uncompilable. This problem was solved by utilizing functions within the makefiles that copied the files to the capitalized versions without disturbing the names of the baseline HP code. There is no such problem with this using g95.

- **C code differences**

Because most of the code in a mixture of C and Fortran, there are cases where the Linux C code requires the call sequence to have an "_" character appended to the name of the module. For example "void mjdymd" becomes "void mjdymd_". It is a consequence of the way that Fortran processes the symbolic names of the called modules. This problem was solved in g95 by using the "-fno-underscoring" compiler option.

There are also a few differences in bounding constants, with values acceptable on the HP that must be modified for Linux.

While compiling with g95 on a 64-bit system, got the following error:

```
/SCF/src/datecon2005/convert.c:
warning: incompatible implicit declaration of built-in function âstrlenâ.
```

Added the following line to convert.c and the warning stopped:

```
long strlen(char *);
```

This fix was able to be included in the HP environment as well.

- **Preprocessor differences**

In the HP code it was acceptable to have commenting placed on the code lines associated with the preprocessor; however, in the Linux code it is not. For example, "#ifdef" cannot appear as "#ifdef ! diagnostic code".

Fortran 90 Code

The Absoft compiler on Linux proved to be far more discerning than the HP compiler. Little code anomalies that were allowed on the HP were incompatible with Absoft. Some of the abnormalities proved to be inaccuracies; however, there were no cases found that made significant execution differences.

Things to note about the Absoft f90 compiler:

- The libraries used under Linux are archived (.a) rather than shared (.sl).
- Makefile compiler options were very different between the two different compilers. In most cases, an equivalent option could be found, although with different syntax.
- Some function modules which were automatically found by the HP compiler needed to be specifically declared for the Absoft compiler.
- The Absoft compiler has an upper limit on the number of continuation lines.

The g95 compiler on Linux proved to be the most discerning compiler of all, probably because it compiles Fortran 95 rather than Fortran 90.

Things to note about the g95 compiler:

- The module names are all in lowercase rather than uppercase.
  i.e. gla12_mod.mod rather than GLA12_MOD.mod
- Compiler options used included: -fendian=big and -fno-underscoring

Common warnings and errors with g95 include the following:

- Warning (158): INTENT(OUT) variable 'i_err' at (1) is never set

  To solve this add the line: i_err = 0

- Error: Operands of comparison operator '.eq.' at (1) are LOGICAL(4)/INTEGER(4)

  To solve this replace: if (BTEST(i_spcm_status,i) == 0) then
  With: if (.NOT.BTEST(i_spcm_status,i)) then

- /dummy/SCF/GSAS/src/elev_lib/e_calcloadtd_mod.f90:184:1: warning: extra tokens at end of #ifdef directive

  To solve this remove any comments on the #ifdef or #endif lines.

- Warning (100): Nonnegative width required in format string at (1)

  To solve this replace: write(hdr_str,fmt="('Recl=',i)") i_reclen
  With: write(hdr_str,fmt="('Recl=',i8)") i_reclen

- Error: Function 'int4' at (1) has no implicit type:

  To solve this replace: call get_anc09_degrades(GLA05%d_UTCTime, INT4(1))
  With:

```
integer(kind=i4b) :: i_choose1or2
i_choose1or2 = 1
call get_anc09_degrades(GLA05%d_UTCTime, i_choose1or2)
```

- During runtime, you may get a message like the following:

  Remaining memory: 5280 bytes allocated at line 169 of
  /SCF/src/common/200611.0/ANC70_mod.f90

To suppress these warnings, you need to put the following line in the .kshrc file:

```
export G95_MEM_SEGMENTS=0
```

- I have sometimes gotten the following error when compiling:

```
GLA01_print_mod.f90:(.text+0x1dcd): undefined reference to
`unsign2sign_mod_MP_unsign2sign_mat_1b'
```

This has always reflected an environment or makefile problem rather than a code problem. One thing of paramount importance is the ordering of the libraries in the make_defs.incl file. They are in reverse order of what-needs-what. i.e. –lplatform needs to be last, everything else uses it. This is the order of the GSAS common libraries in /SCF/cc_util/make_defs.incl:

```
LIBS=  -lscfcommon \
       -lexec -lprod -lanc -lfile -ltime \
       -lgeo -lmath -lerr -lcntrl -lplatform
```

Other bugs found with the g95 compiler:
- Need to “deallocate” every allocated array.
- “Rewind” statement gave run-time error.
- The “back” option in n=index(val, ' ', back) is outdated.
- Float rounding is different so comparison checking has to be careful.

Spawning Unix Commands

Problems were encountered where Unix commands were spawned within the code. Examples include the use of “wc”, “sort”, and “grep”. These commands had different default settings and flag options. Also, the number of spaces between printed values was greater on the Linux system. This created problems when the software would parse a string based on a single space.

IDL

Very few problems were encountered running IDL, except for the Big Endian/Little Endian incompatibility between the systems. The HP UX stores data in Big Endian format while our Linux system stores data in Little Endian format. Therefore, whenever we opened a binary file to read, we had to include the following syntax:

```
openr,lun,filename,/get_lun,/SWAP_IF_LITTLE_ENDIAN
```

There has been a problem reported on some discussion boards about IDL crashing with a segmentation fault on certain versions of Linux when using display functions like “tv” or “device”.

http://www.ittvis.com/forum/message.asp?fmid=2267

Apparently IDL 6.3 has problems with the newest versions of the X11 libraries. You might run into it in either Fedora Core 5 or 6. It's supposed to be fixed under IDL 6.4.

http://www.ittvis.com/services/techtip.asp?ttid=4177

System

The hostname on the linux systems includes the full system name while the HP hostname is just the first part:

i.e. linux hostname: rscf.colorado.edu; HP hostname: rscf

Appendix A: Creating SSH Keys for Non-Interactive Data Exchange

This is the configuration of a user account on a UNIX system to permit non-interactive SSH based data transfers between two UNIX systems. This approach will utilize public-key authentication with plain text key files (no passphrase). This procedure is For SSH protocol version 2.

1) Permissions on /SCF need to be drwxr_xr_x (chmod 755) (ls -ld to see)

2) Create the /SCF/.ssh directory if not there. The permissions are drwx_____ (chmod 700)

3) On the remote site, generate the DSA authentication key pair in /SCF/.ssh:
   - /usr/bin/ssh-keygen –t dsa –N “”

4) Copy /SCF/.ssh/id_dsa.pub from the remote site to the main site. Make sure to rename id_dsa.pub to id_dsa.pub_<rscf> on the main site or you will overwrite the main site’s id_dsa.pub file.
   - i.e. scp /SCF/.ssh/id_dsa.pub icesat0:/SCF/.ssh/id_dsa.pub_glo

5) Copy /SCF/.ssh/id_dsa.pub from the main site to the remote site. Make sure to rename id_dsa.pub to id_dsa.pub_<mscf> on the remote site or you will overwrite the remote site’s id_dsa.pub file.
   - i.e. scp /SCF/.ssh/id_dsa.pub glo:/SCF/.ssh/id_dsa.pub_icesat0

6) On the remote site, concatenate /SCF/.ssh/id_dsa.pub_<mscf> to the end of the authorized_keys file. If the authorized_keys file doesn’t exist, then copy id_dsa.pub_<mscf> to authorized_keys.

7) On the main site, concatenate /SCF/.ssh/id_dsa.pub_<rscf> to the end of the authorized_keys file. You can either cut and paste the files or cat them.

8) Have Steve Fiegles add the abbreviated host name (i.e. hudson for hudson.geology.buffalo.edu) to /etc/hosts on icesat0.

9) Permissions for files under /SCF/.ssh are rw_____ (chmod 600)

10) Test the new keys
    - Test the ability to ssh from icesat0 to the new system and back to icesat0. There should be no need to type in a password; if there is a password prompt, then the keys don’t work. You may be prompted with a “yes/no” question the first time you ssh in; that’s normal – once you type “yes”, you should never be prompted again.
    - Test the ability to scp a small file from icesat0 to the new system and back to icesat0. Again, there should be no password prompts and answer “yes” to the “yes/no” question.

Note: if you experience problems, check the permissions on the .ssh directory and files below. They are usually very restrictive (i.e. rw by scf only) and if they are more open, may not work.

Appendix B: Installing mysqltcl for sites that access a MySQL database

Here's how to install mysqltcl on a 32-bit Linux system:

1) go to http://www.xdobry.de/

2) download the mysqltcl package mysqltcl-3.03-0.i386.rpm

3) install it

Here's how to install mysqltcl on a 64-bit Redhat Enterprise 4.0 Linux system:

1) go to http://www.xdobry.de/

2) download an uncompress the mysqltcl package mysql-3.03.tar.gz

3) edit the "configure" file, changing the line:
   "'ls -d /usr/lib 2>/dev/null \'" to "'ls -d /usr/lib64 2>/dev/null' \"

4) Issue the following commands:
   - ./configure --with-mysql-lib=/usr/lib64/mysql
   - make
   - make install

Appendix C: Recommendations on Selecting a Linux Operating System

An “ideal” Linux distribution should have several desirable characteristics:

1) Long term stability based upon software and security updates for at least 5 years (an "enterprise" operating system).

2) A diverse, rich software catalogue which can be maintained within the resident package management system (yum, apt, up2date, etc.).

3) Possibility of operating system upgrades (from major version to major version) without significant downtime.

Assessments of some Linux distributions:

Fedora Core - large , diverse software catalogue within package management system but an unacceptably short support lifecycle (~ 1.5 yrs), free

Red HatEnterprise Linux (RHEL) - good long term stability (5 yrs or more), commercial RED HAT support, relatively limited software catalogue within package management system (3700 packages), yearly maintenance: $150-250

Centos 5.1 - RHEL 5 codebase, good long term stability (5 yrs or more), relatively limited software catalogue (2700 packages) within package management system, free

Ubuntu 8.04 LTS Server- good long term stability (5 yrs or more), large , diverse software catalogue within package management system (25000 packages), free, possible ability to upgrade between major versions with relative ease (a Debian linux trait)

At this point in time, we believe that Ubuntu 8.04 LTS holds the greatest promise as the Linux distribution of choice. We have successfully installed and tested the SCF software on an Ubuntu test system.