Advanced SUP Use

The SUP virtual file system (VFS) capability can be used to issue file-related commands to all SUP-connected hosts from the command line of your local host.

Accessing Files Across Multiple Hosts

The SUP client includes a virtual file system capability that allows files across all SUP-connected resources to be accessed using standard file system commands. For example, once you have activated VFS as described below, the command:

% ls pfe*21*:/tmp

from your local host would list the files in /tmp on pfe*21*. The command:

% cp *filename* pfe*21*:/tmp

from your local host would copy the file *filename* from your current directory on your local host to /tmp on pfe*21*.

The set of supported commands includes cat, cd, chgrp, chmod, chown, cmp, cp, df, diff, du, file, grep, head, less, ln, ls, mkdir, more, mv, pwd, rm, rmdir, tail, tee, test, touch, and wc.

Note that this functionality is not a true file system since only these commands are supported and only when used from within a shell. Unlike more general approaches such as FUSE, however, the SUP capability is completely portable and can be enabled with no additional privileges or software.

Commands through the VFS functionality can act on any combination of local and remote files, where remote files are prefixed with *hostname*:. For example, the command:

% cat pfe*21*:/tmp/*rfile* ~/*lfile*

would print the file *rfile* in /tmp on pfe*21* as well as the file *lfile* in the user's home directory on the local host to the terminal. Any number of hosts can be included in any command. For example, the command:

% diff lfe2:/tmp/*lfe_file* pfe*21*:/tmp/*pfe_file*

would show the differences between the file *lfe_file* in /tmp on lfe2 and the file *pfe_file* in /tmp on pfe*21*. The client determines if any remote access is needed based on the path(s) given. If not, it will execute the command locally as given as rapidly as possible. Fully local commands also support all options with the exception of options of the form -f *value* (that is, single-dash options that take values).

VFS Activation

Requirements

Currently, SUP VFS functionality is only supported for bash, but csh support is planned for the future. This functionality requires Perl version 5.8.5 (note that this is more recent than version 5.6.1 required by the basic client functionality). It also requires the standard Unix utilities cat, column, false, sort, and true; and has been tested successfully on Linux, OS X, and Windows underCygwin. Note that users of Windows under Cygwin may need to install the coreutils and util-linux packages to obtain these utilities.

Activation and Deactivation on Your Local Host

Complete these steps.

1. Install the SUP client if you have not already done so.
2. Activate VFS functionality in a bash shell.
    - For an interactive bash shell, run:

      ```
      eval `sup -s bash`
      ```

    - For a non-interactive bash shell, begin the script with the following:

      ```
      #!/bin/bash
      shopt -s expand_aliases
      eval `sup -s bash`
      ```

    The instructions will load aliases and functions that will intercept specific commands and replace them with commands that will perform the requested actions through the SUP client.

To deactivate VFS functionality in a bash shell or bash script at any time, run:

% eval `sup -r bash`

Command-Line Options

The behavior of the virtual file system can be modified using various options at the time it is activated.

-ocmd=opts
Specify default options for a given command since the VFS functionality overrides any existing aliases for its supported set of commands.

-t *transport*
Change the file transport from its sftp default to transport. Currently, the only additional transport available is bbftp. Note that using bbftp as the transport may slow down certain operations on small files as bbftp has higher startup overhead.

-u *user*
Specify NAS user name. Note that this option is *required* if your local user name differs from your NAS user name.

For example, the following invocation activates the client virtual file system using bbftp as the transport mechanism, *nasuser* as the user and adds colorization of local file listings using the Linux ls --color=always option.

```
% eval `sup -s bash -t bbftp -u nasuser -ols=--color=always`
```

VFS Caveats

The VFS functionality is still somewhat experimental. In general, it works for the most common usage scenarios with some caveats. In particular:

- "Whole file" commands (that is, commands that must process the entire file), including cat, cmp, diff, grep, wc (and currently more/less due to implementation) retrieve files first before processing for efficiency. Thus, these commands should not be executed on very large files.
- There is a conflict between commands that take piped input and the custom globbing of the client, thus these commands have portions of globbing support disabled. These commands are grep, head, less, more, tail, tee, and wc. In these cases, globbing will work for absolute prefixes, but not relative. For example, grep *filename* pfe*21:*/tmp/* will work, but cd pfe*21:*/tmp; grep *filename* * will not.
- Redirection to/from remote files doesn't work. The same effect can be achieved using cat and tee (for example, grep localhost < pfe*21:*/etc/hosts > pfe*21:*/tmp/a could be done with cat pfe*21:*/etc/hosts | grep localhost | tee pfe*21:*/tmp/a > /dev/null). Redirection still works normally for local files.
- The first time a command is run involving a particular host, a SFTP connection is created to that host. When runningps, it may appear as if a zombie client process is running.

VFS Commands

Currently supported commands and their currently supported options are below. Unsupported options will simply be ignored except where noted. All commands are still subject to SUP authorizations, thus something that cannot be executed or written normally through the SUP cannot be executed or written through this functionality either.

- cat (no options)
- cd (no options)
  Note that when changing to remote directories, cd only changes $PWD, so to make changes visible the working directory (that is, \w in bash) must be in your prompt. For example, the following prompt:

  ```
  export PS1="\h[\w]> "
  ```

  would display the current host name followed by the current working directory.
- chgrp (no options)
  Groups may be specified either by number or by name. Names will be resolved on the remote host.
- chmod (no options)
  Modes must be specified numerically (for example, 0700). Symbolic modes, such as a+rX, are not currently supported.
- chown (no options)
  Users and groups may be specified either by number or by name. Names will be resolved on the remote host.
- cmp (all options)
- cp [-r]
  Note that copies between two remote hosts transfer files to the local host first since the SUP does not allow third party transfers. Thus, very large file transfers between remote systems should be achieved using an alternate approach.
- df [-i]
  Note that 1024-byte blocks are used.
- diff (all options)
- du [-a] [-b] [-s]
  Note that 1024-byte blocks are used.
- file (all options)
- grep (all options)
- head [-*number*]
  Note that head does not support the form -n *number*, so, for example, to display the first 5 lines of a file, use -5 and not -n 5.
- less (all options)
- ln [-s]
  Note that hard links are not supported. Links from remote files to local files (for example, ln -s pfe*21:*/*filename* /*filename*) will be dereferenced during certain operations (for example, cat /*filename* will cat pfe*21:*/*filename*).
- ls [-1] [-d] [-l]
  For efficiency purposes, ls behaves slightly differently for remote commands than for local commands. In particular ls -l will not

show links by default and will show what is actually linked instead of the link itself. Link details can be obtained using the -d option (for example, ls -ld *).

Also for efficiency, ls processes remote files before local files, so output ordering may be changed when remote and local files are interleaved on the ls command line. For example, ls /*file1* pfe*21*: /*file2* would show pfe*21*: first, then /*file1*, then /*file2*.

- mkdir (no options)
- more (all options)
- mv (no options)
- pwd (no options)
- rm [-r]
- rmdir (no options)
- tail [-*number*]
  Note that tail does not support the form -n *number*, so, for example, to display the last 5 lines of a file, use -5 and not -n 5.
- tee [-a]
- test [-b] [-c] [-d] [-e] [-f] [-g] [-h] [-k] [-L] [-p] [-r] [-s] [-S] [-u] [-w]
  Note that compound and string tests are not supported. Compound and string tests can be achieved using multiple test commands separated by shell compound operators. For example, instead of

  % test -f pfe*21*:/*filename* -a "*abc*" != "*123*"

  do

  % test -f pfe*21*:/*filename* && test "*abc*" != "*123*"

  Alternatively, the actual test command can be executed through the SUP:

  % sup ssh pfe*21* test -f /*filename* -a "*abc*" != "*123*"

- touch (no options)
- wc (all options)