The Improved “lfs find” Command

Rick Mohr, Anjus George, and James Simmons

National Center for Computational Sciences

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle LLC for the US Department of Energy

Overview

- LU-10378 and the need for a `--printf` option
- New format directives
- Simple examples
- Overview of code changes for implementation
- Future work and potential improvements

LU-10378: ”lfs find” is missing ”-printf” support

- Would be useful if `lfs find` had a `-ls` option
- Even better, add a `--printf` option and have the `-ls` option be implemented as a fixed format of `--printf`
- For info that is not Lustre-specific, try to use the same % format directives as `find`
- For Lustre-specific info, create new directives prefixed with `%L` (ex - `%Lc` for stripe count)

Search Options

- `lfs find` search uses many of the same options as `find`
  - atime/mtime/ctime
  - UID/GID
  - File size and type
  - Etc.
- Also includes a lot of Lustre-specific search options
  - Component count
  - Mirror state
  - Stripe size/count
  - OST index
  - And many more

Supported Actions

- `find`
  - -exec
  - -execdir
  - -ok
  - -ls
  - -print
  - -printf
  - -print0
  - -fls
  - -fprint
  - -fprintf
  - -fprint0

- `lfs find`
  - --print
  - --print0

Format Options

- The new `--printf` option for `lfs find` is intended to support several of the same ‘\’ escapes and ‘%’ directives as `find`

\n	Newline
\t	Horizontal tab
\\	Literal backslash
%%	Literal percent sign
%a, %c, %t, %w	Timestamps in ctime format
%A@, %C@, %T@, %W@	Timestamps in seconds since epoch

%b	Disk space used (in 512-byte blocks)
%G	Numeric group ID
%m	File permission bits (octal)
%p	File’s name
%s	File size (bytes)
%U	Numeric User ID
%y	File’s type

Note: %w and %W@ (file birth times) are not standard with `find`

Lustre-specific format options for `--printf`

- New ‘%’ directives were added to print Lustre-specific details

%Lc	File’s stripe count
%LF	File Identifier (FID) associated with file
%Lh	Directory’s hash type
%Li	File’s starting OST index (or starting MDT index for directory)

%Lo	List of all OST/MDT indices associated with file or directory
%Lp	File’s OST pool
%LP	Numeric project ID
%LS	File’s stripe size

Note: For composite files, the %Lc, %Li, %Lp, and %LS directives will use info from the last instantiated component.

Examples

```
#> lfs setstripe -E 10M -c 2 -S 1M -E eof -c 8 -S 4M data.txt
#> dd if=/dev/urandom of=data.txt bs=1M count=100
#> lfs find . -type f --printf ”path = %p\n stripe_cnt = %Lc\n stripe_size = %LS\n
start idx = %Li\n %Lo\n”

path = ./data.txt
 stripe_cnt = 8
 stripe_size = 4194304
 start idx = 4
 [9,0][4,3,1,7,10,8,6,12]


#> lfs find . -type f -size +100G --printf ”%p (%Lc)\n”
./file4 (16)
./file1 (1)


#> lfs find . -type f -size -20M --printf ”%p (%Lc)\n”
./file5 (16)
./file2 (2)
```

Search for improperly striped files.

Code Changes

- A few changes are added to `lustre/utils/lfs.c` for processing the new `--printf` option
- In `lustre/include/lustre/lustreapi.h`, a couple of new fields are added to `struct find_param`
- Most changes are in `lustre/utils/liblustreapi.c`
  - Vast majority of the new code deals with parsing the printf string and formatting the output
  - Some changes were made to the `cb_find_init( )` function to alter the flow of how file metadata was gathered because…

metadata needed for searching != metadata needed for printing

Code Changes (cont.)

Before:

```
if (param->fp_check_projid) {
        fd = open(path, O_RDONLY);
        ret = fget_projid(fd, &projid);
        if (projid != param->fp_projid) {
                goto decided;
        }
}
```

Only retrieve the project ID if we need that info for the search criteria

After:

If `--printf` option is given, set the `gather_all` flag and force collection of information

```
if (param->fp_check_projid || gather_all) {
        ...
}
```

Warnings and Caveats

- `lfs find` will not work on non-Lustre files if `--printf` option is used
- Can be additional load on the MDS servers if `--printf` is used and all files are matched
  - Ex: lfs find /lustre –printf "%p %s %Lc\n"
- Recently discovered odd behavior with some edge cases
  - Mostly related to %Lo with non-instantiated components
- Good News: without the `--printf` option, original code paths should remain unchanged

Future Work / Improvements

- Other potential format directives
  - %u / %g for user / group name
  - Print component flags for composite files
- Improve efficiency of printf string parsing/formatting
  - Currently reparses string on every matched file
  - Perform single pass and tokenize the string for reuse
- Be more selective in gathering metadata
  - Choose what to collect based on directives in printf string
- Add/improve test cases in sanity script
- **Found a bug? Please file a Jira ticket**

Acknowledgments

- Implementation was done in conjunction with Anjus George
- Thanks to James Simmons for helping to explain some of the black magic inside the Lustre code
- Thanks to Andreas Dilger for the detailed code reviews (and patience with our many iterations)

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by The Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Questions?