

高速ファイル転送ソフト SRFT

情報基盤センター

松葉 浩也

1 はじめに

本センターに導入されている SR11000 では、ファイルシステムとして主に HSFS が使用されています。このファイルシステムは最適化されたプログラムが大量のデータを扱うときには高い性能を発揮しますが、HSFS の特性を考慮していない一般のプログラムでは本来の性能は得られません。ファイル転送に使用する SCP は HSFS の特性を考慮していないプログラムであるため、ファイルアクセスの性能が悪く、大容量ファイルの転送に長い時間を要します。この問題を解決するために HSFS に最適化したファイル転送ソフト SRFT を作成しました。実験環境において SRFT は SCP の 26 倍である 70MB/s を達成しています。

2 インストール

最初にお使いになる前に、本稿の最後にある注意事項をよくお読みください。
SRFT は SR11000 と手元のクライアント双方にインストール作業が必要です。

SR11000

SR11000 には専用のバイナリを準備しました。コンパイル作業は不要です。

1. <http://www.il.cc.u-tokyo.ac.jp/~matsuba/srft/srft-1.1.0-sr11000-bin.tar.gz> をダウンロードする
2. 上記ファイルを scp など通常使用している方法で SR11000 にコピーする
3. `$ gzip -dc srft-1.1.0-sr11000-bin.tar.gz | tar xvf -`
で解凍する。./local/bin の中に必要なファイルが展開される。
4. 展開されたファイルが存在するディレクトリに PATH を通す(必須)。手順 3 の作業をホームディレクトリで行った場合は、~/cshrc に以下のように書く。
`set path=(/batch/xxxxx/local/bin $path)`

ソースからのコンパイルも可能ですが、そのためには OpenSSL もコンパイルしてインストールする必要があります。

手元のマシン

対応 OS は Linux, FreeBSD, MacOS X です。Windows は Cygwin も含め非対応です。また、OpenSSL の開発環境が必要です。通常 openssl-devel のようなパッケージ名で提供されているはずですが。

1. `$ wget http://www.il.cc.u-tokyo.ac.jp/~matsuba/srft/srft-1.1.0.tar.gz`
2. `$ tar zxvf srft-1.1.0.tar.gz`
3. `$./configure --prefix=$HOME/local`
OpenSSL を特殊なディレクトリにインストールした場合は、`./configure` のオプションに `--with-openssl-prefix=<openssl の path>` を加えてください。
4. `$ make`
5. `$ make install`
6. インストール先に PATH を通す。tsch のときは上記 SR11000 と同じ方法、bash の時は `~/bashrc` に以下のように書く。
`export PATH= /local/bin:$PATH`

3 使いかた

なるだけ scp に近い方法で使えるように作りました。

- 単一ファイルの転送
`$ srft ~/file.dat xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:`
- 複数ファイルの転送
`$ srft ~/file.dat ~/somefile*.dat xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:`
- SR11000 からのダウンロード (1)
`$ srft xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:file.dat .`
- SR11000 からのダウンロード (2)
`$ srft xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:file*.dat .`
- ディレクトリごと転送 (1)
`$ srft -r ~/dir1 xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:somedir/`
- ディレクトリごと転送 (2)
`$ srft -r xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:somedir/ .`

4 起こりがちなエラー

- `srft_server: Command not found.`
リモートホストで PATH を通す作業ができていません。インストール方法に書いてある方法で PATH を通してください。
- `Error: reading public key: xxxxxx`
認証に使っている鍵が見つかりません。rm -r `~/srft` を手元のクライアントと SR11000 の両方で実行

してから再度試してみてください。

5 性能

以下に SRFT の実行結果と、SCP の実行結果を載せます。x1 x2 は 500MB のファイルです。実験に使用したクライアントは Intel Xeon Processor 5140 x2 のマシンに Hitachi SANRISE AMS200 を接続したコンピュータです。また、ネットワーク的には SR11000 と近く Gigabit Ethernet の帯域が十分に発揮できる環境です。この実験環境では SCP の 2.7MB/s に対し、SRFT は 26 倍の 70.4MB/s を達成しています。ディスク、ネットワーク共に理想的な環境での測定なので、他の環境ではここに示すほどの性能は出ないことが予想されます。

SRFT

```
[matsuba@fs ~]$ srft x1 x2 xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:
xxxxxx@sr11000-s.cc.u-tokyo.ac.jp's password:
x1                500.0MB/500.0MB (100.0%) 72.4MB/s 0h00m07s
x2                500.0MB/500.0MB (100.0%) 73.3MB/s 0h00m07s
Total             1.0GB/ 1.0GB (100.0%) 70.4MB/s 0h00m14s
```

Total 性能が各ファイルの転送性能の平均値を下回るのは、Total にはどのファイルの転送にも属さない制御情報の交換時間が含まれるためです。

SCP

```
[matsuba@fs ~]$ scp x1 x2 xxxxxx@sr11000-s.cc.u-tokyo.ac.jp:
xxxxxx@sr11000-s.cc.u-tokyo.ac.jp's password:
x1                100% 477MB 2.7MB/s 02:59
x2                100% 477MB 2.7MB/s 02:59
```

6 技術的情報

6.1 安全性

転送経路中、データは RC4 と呼ばれる簡単な方式で暗号化されています。途中でパケットを盗聴されてもすぐに内容がわかることはありません。ただし、暗号強度はそれほど強くないので、何者かが悪意を持って解析すれば内容が判明してしまうかもしれません。一方、通信相手の正しさ（本当に SR11000 と通信しているかどうか）に関しては、最高で SSH と同等の安全性ということになります。これは SSH を用いて証明書の交換を行っているためです。ネットワーク上での改竄は検出できるはずですが、

6.2 高速な理由

HSFS で高速にファイルアクセスを行うためには、read または write システムコールに渡すバッファのサイズを 16MB 程度にする必要があります [1]。一方 Linux などで使用されている PC 向けファイルシステムでは 4KB 程度のバッファで十分な性能が出るので、SCP を含めほとんどのアプリケーションは 4KB 程度の単位でシステムコールを行います。そのため、これらの一般的なアプリケーションを HSFS で使用しても十分な性能が得られないこととなります。SRFT は 16MB 単位でファイル入出力を行うよう、送受信データをバッファリングしています。このバッファサイズの最適化の他、ファイルアクセス待ちの時間を暗号化処理に活用するなどの工夫を行い、上記の性能を実現しています。

7 Q & A

- 80MB/s に程遠い性能しか出ません
途中で 100Mbps のネットワークが挟まっていないことを確認して下さい。また、上述のように SRFT では 16MB 単位でアクセスを行うことで高速化しているので、16MB に満たない小さなファイルが大量にある場合は SCP と同等の性能しか出ません。
- うまく動かない / ファイルが壊れる
まずは表示をよく見て "Permission denied" や "No space left on device" などの分かりやすいエラーが出ていないことを確認してください。バグが疑われる時は状況を作者までご連絡ください。

8 注意事項

以下の事項を了解の上ご使用下さい。

- 本ソフトを使用したことによるいかなる結果に対しても、作者の松葉および情報基盤センターは一切の責任を負いません。
- 本ソフトは情報基盤センターによるスーパーコンピュータサービスの一部ではありません。情報基盤センターとしては本ソフトに関する質問はお受けできません。質問、要望などはシステムの相談窓口でなく、直接松葉までお願いいたします。ただし、作者もサポート義務は負いません。
- 本ソフトのライセンスは GPLv2 です。

9 最新情報および連絡先

松葉 浩也

本ソフトの URL: <http://www.il.cc.u-tokyo.ac.jp/~matsuba/srft/srft.html>

参考文献

- [1] 黒田 久泰 "C 言語におけるファイル入出力の高速化", スーパーコンピューティングニュース Vol.8 No.5 pp.28-33, 2006 年 9 月