

株式会社 ITS MORE

2020年4月設立

`scp f1 f2 ≐ rsync f1 f2`

2020年5月9日 投稿者: YSATO@DELEGATE.ORG

rsync 最高っす！

開発しているプログラムを別のプラットフォーム上でコンパイルして実行してテストしたい。少し変えてはテストしたいので、コンパイル結果は10秒程度で得られると良い。手持ちのツールだけでやりたい。

そう考えたのでとりあえず、変更したらプログラムをリモートのマシンにscpして、sshでコンパイルやテストのコマンドを送るという安直な方法にしました。今日のOSはデフォでsshを装備しているので、追加でインストールするものではありません。

この手順を行う簡単なシェルスクリプトを書いただけで実現。変更分だけの再コンパイルであれば、差分が小さければ、コマンド打って10秒程度で結果が得られるのでほぼ満足しています。

気になったのは、プログラムをリモートに送り込むための時間です。ソースコード類のサイズはナマだと約10MB、tarしてgzipして2.5MB。scpだと状況によっては転送に数秒かかります。それだけでなく、これを何百回もやっていると、サーバ側の従量ネットワーク課金が気になってしまいます。

それで思い出すのが昔懐かしい `rsync` です。ググってみると最近は通信に `ssh` を使うようになっていて、すごく健在のようでした。やはり原理的に良いものは生き残るのだな。よし、これで行こう。手元のマシンでも、リモートでも `rsync` とコマンドを打つと何か動くので、インストールされているようです。

どう使うのか？ `rsync -h` とかするとどどっと出てくるので読む気にならない（（実は `man rsync` の先頭だけ見ればサクッとわかることに後で気づく））。`scp` と同じノリで使えるとラッキーと思い、

```
% rsync ファイル リモートホスト:
```

としたら、そのまま動きました。素晴らしいです。

`scp f1 f2 ≐ rsync f1 f2`

性能的にはどうなのでしょう？超簡単なベンチマークをやってみました。

```
% openssl rand 10000000 > 10MB
% time rsync 10MB remote:
rsync 10MB remote: 0.46s user 0.37s system 6% cpu 12.435 total
% time rsync 10MB remote:
rsync 10MB remote: 0.06s user 0.08s system 7% cpu 1.833 total
% time rsync 10MB remote:
rsync 10MB remote: 0.06s user 0.07s system 7% cpu 1.804 total
```

リモートのファイルの `cksum` を確認しましたが、もちろん同一です。そしてこの例では、2回目以降は何もデータ内容に変更が無いので、何も送られていないようです。macOSの（とてもステキな）アクティビティモニターには、一発目の転送しか出てきません。

受信パケット数:	6,219,309	データ ↕	受信したデータ:	2.44 GB
送信パケット数:	7,466,336		送信したデータ:	4.44 GB
毎秒受信パケット数:	14		毎秒データ受信:	3.89 KB
毎秒送信パケット数:	14		毎秒データ送信:	2.00 KB

rsyncでのデータ転送(差分のみ)

一方 scp では毎回データを転送してますので、毎回ニョキニョキと送信データグラフが伸びます。



scpでのデータ転送 (毎回全転送)

```
% openssl rand 10000000 > 10MB
% time scp 10MB remote:
10MB 100% 9766KB 796.4KB/s 00:12
scp 10MB remote: 0.18s user 0.20s system 2% cpu 13.473 total
% time scp 10MB remote:
10MB 100% 9766KB 893.0KB/s 00:10
scp 10MB remote: 0.19s user 0.20s system 3% cpu 12.118 total
% time scp 10MB remote:
10MB 100% 9766KB 497.7KB/s 00:19
scp 10MB remote: 0.19s user 0.19s system 1% cpu 20.791 total
...
```

このホスト関係、ネット環境だと、ssh 接続の確立に1.2秒かかる（この遅延、ログインする時に結構気になるんですが）ようなので、それよりは短縮できないですね。常設のトンネルw開通させておくしか無い。rsync自体の空作業は0.1秒未満の様様。

```
% echo a > a
% time scp a remote:
a 100% 2 0.0KB/s 00:00
scp a remote: 0.01s user 0.01s system 2% cpu 1.217 total

% echo b > b
% time rsync b remote:
rsync b remote: 0.01s user 0.02s system 2% cpu 1.283 total
```

考え所は、クラウド上のサーバのデータ（主に差分はログだけ）のバックアップも rsync でやるかどうかです。ファイル数が70万くらいあるので、変更されたファイ

ルを効率的に知る方法が無いと、それを探索し回るのに結構な負荷がかかりそうです。まあ、変更した本人のサーバかOSが教えてやればよいのでしょうか。

とりあえずは、サーバ仮想マシンのディスクのスナップショットをとるほうが簡単だし全面的ですし、スナップショット保管料金は1日4円程度です。なので、あれはあれで良いかなという感じ。

クラウド上のドライブサービスも、rsync的な仕組みが入っていると良いですね。入ってるのかな？

2020-0509 sato@izmoh

rsync-最高っす！ - 株式会社-ITS-more

