

Optimizing scp and rsync/ssh to use high bandwidth connections

Motivation

SSH is often used to transfer files (or streams), by `scp` (always) and often by `rsync` (if used over `ssh`, i.e. with option `--rsh ssh`).

By default `ssh` as well as `rsync` tend to try to compress the data transferred. This is useful over slow connections, like a typical SOHO internet line, especially when using the slow upload channel of asynchronous technologies or plans.

If the network connection is fast, as in a GBit connection within a LAN or DMZ, even with top-notch CPUs those compressions become bottlenecks and reduce the transferspeed from say 100 MB/s to 25 MB/s or even less.

Therefore it sometimes makes a lot of sense and saves a lot of time to suppress those compression attempts.

Suppressing ssh's compression

To suppress `ssh`'s compression for a specific `scp` call, add `-o 'Compression no'`.

To generally suppress `ssh`'s compression for connections to certain hosts create or adapt Host section(s) in `/etc/ssh/ssh_config` or `~/.ssh/config`, like this:

```
Host *.dmz.foobar
Compression no
```

This applies to all `ssh` connections to those host(s), whether you use `ssh`, `scp`, or `rsync` over `ssh`

Suppressing rsync's compression

If calling `rsync` directly, do NOT use `--compress` (resp. `-z`).

Some `rsync` wrappers like our `xsync` script add (i.e. "inject") `--compress` for remote connections automatically, but allow to pass through further `rsync` options. In that case add `--compress-level=0`.

Special note for rsync over ssh

If using `rsync` over `ssh`, both tools's compression attempts must be avoided separately!