株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.08.05 22:26:14 +09'00'

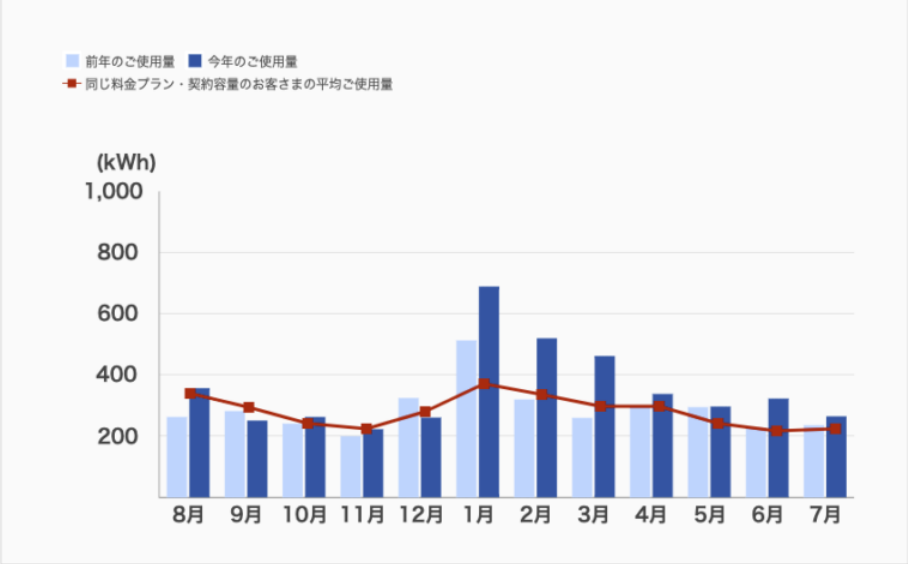
2020年8月5日 投稿者: SATOXITS

scpを驚速化するソフトの開発に成功

開発：なんか変な時間から始業になりました。

社長：涼しい夜に仕事をするというのは合理的かと思います。

経理：そういえばでんこちゃんの家計簿が見れるようになりました。



基盤：この赤い線、世間一般世帯の平均ということだと思いますが、それにとても近いのが驚きですね。

社長：うちと違って一般家庭では夏場は冷房を入れると思うんですが、大したピークにも見えないのが不思議です。

＊　＊　＊

開発：さて、問題のscpのアップロードですが、scp.c の中の source という関数の中のループで実行されています。ですが、そのあたりは 7.7と7.8で違わない。そこから呼ばれて実際に送信を行っているatomicio6という関数もです。

開発：で、scpがどこに出力をしているかなのですが、サーバへのコネクションを張ったsshに違いないと思うわけです。実際、転送中のプロセスはこんな形になっています。7.7と7.8で違わない。

```
 1045 pts/2    Ss     0:00  \_ -bash
23278 pts/2    S+     0:00      \_ ./scp /home/ysato/100MB dg9:
23279 pts/2    S+     0:00          \_ /home/ysato/openssh/openssh-7.8p1/installed/bin/ssh
-x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -oRemoteCommand=none
 -oRequestTTY=no -- dg9 scp -t .
```

```
 6645 pts/4    Ss     0:00  \_ -bash
24469 pts/4    S+     0:00      \_ ./scp /home/ysato/100MB dg9:
24470 pts/4    S+     0:00          \_ /home/ysato/openssh/openssh-7.7p1/installed/bin/ssh
-x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -oRemoteCommand=none
 -oRequestTTY=no -- dg9 scp -t .
```

開発：それでは、ssh は何をしているかというと、ssh.c の ssh_session2 から clientloop.c:client_loopを呼んでいて、client_loopではselectで待っている。何を待っているかと言えば、やはりサーバへの送信が可能になっているのを待っているのかなと思うわけです。

社長：pollじゃなくてselectなのがBSD的なんでしょうかね。私も昔はselectでしたが。pollのほうが使いやすかったので、すっかりpollになりました。

開発：しかし人の書いたプログラムの動作を追うには、ブレークポイント機能のあるIDEがないと手間ががかかるというのをつくづく感じますね。まあマルチプロセスだとちょっと厄介ですが、scp | ssh のこれとか、別にスレッドで良いと思うんですが。そのほうが性能も多少良いかも知れないし。

開発：で、output のselectの対象になってるサーバ向けのソケットのバッファサイズは87KBあるので、問題ないと思われます。そもそも 7.7 と同等でしょうし。実際これを512KBとかにしてみても、変化はありません。

社長：なるほど。となると、どこかでしょうもない遅延が入ってるんでしょうね。わくわく。

＊　＊　＊

社長：ところで Vivaldi をメインブラウザにしてから2ヶ月になります。操作感的な面白さはやや飽きたというか、良し悪しな面もあるように思うようになりました。基本機能的なところで完成度が今ひとつに感じるところもあります。ですが、やはりVivaldiがいいなと思う点は、複数の別個のユーザとしてプロファイルを定義できて、使い分けられる所かなと思います。そしてセッションの保存と回復。現在開いているウィンドウ・タブのセットを名前付きで保存して、回復できる。これはまさに、昔からずっと欲しかった機能なわけです。

社長：ちょっと残念なのは、複数のセッションを並列で持てないように見える点。あと、セッションをまるごと削除できないようにみえる点です。あるいは、各セッションに含まれるウィンドウとタブを一覧表示する機能が無いように見えること。タスクウィンドウみたいので良いと思うのですが。

社長：Vivaldiには、ブラウザを使うにあたって、あるユーザとして、あるセッションの中で、あるウィンドウを開いているのだ、あるいは開くのだという使い方を、もっと明確に押し出してい欲しいと思いますね。だから、使い始める時に「デフォルトのユーザ」として「デフォルトのセッション」の中に居るのだということをわかりやすくしてほしい。あるいは「保存されたセッション」に「デフォルトのセッション」が無いのも片手落ちと思います。これはもやもやっとしているし、勝手に回復することになってい

るのが良いとも限りません。

基盤：Person1とか英語としてもイマイチですが、日本語版でもなにか工夫して欲しいものです。

開発：日本語のソフトだと「既定ユーザ」とかが好まれそうなｗ

社長：あとは複数のユーザ（プロファイル）の間でセッションというかウィンドウやブックマークを共有出来ないのも困ることがあります。たとえばこのセッションやブックマークはどのユーザと共有する、という設定ができると良いと思います。そしていずれかはそれらを、別種のブラウザとの間でも共有できると良いのですが。

開発：まあ抱き合せ機能で囲い込み合戦中でしょうからね。そういう共有のための標準化がされてしまった後に、Vivaldi独自の強みというのを維持し続けられるかも問題かなと思います。

社長：個別には実現技術的に難しい話ではないでしょうしね。

基盤：カスタマイズ能力を標榜するなら、カスタマイズを支援するウィザード的なものも充実してほしいですよね。沢山の設定項目の中でどれが自分の興味のあるものかとか。あるいは、プレカスタマイズされたテンプレート的なものを提供するとか。カスタマイズ情報のエクスポートとインポートって出来るんですかね？

社長：だれそれ監修の設定セットとかテーマとか、設定に名前を付けられるようになると良いですね。

＊　＊　＊

開発：さてそれで、selectにラッパーをかけて、動作を観察しました。selectに10ms以上かかった場合だけ出力しています。まず 低速な7.8の場合。

```
[18202] clientloop.c:577 select(8,59)=1 49972 us R[3]
[18202] clientloop.c:577 select(8,59)=1 13248 us R[3]
100MB                              0%    0     0.0KB/s   --:-- ETA[
18202] clientloop.c:577 select(8,59)=1 45140 us R[3]
[18202] clientloop.c:577 select(8,59)=1 42863 us W[3]
[18202] clientloop.c:577 select(8,59)=1 65321 us R[3]
[18202] clientloop.c:577 select(8,59)=1 45639 us W[3]
[18202] clientloop.c:577 select(8,59)=1 54407 us R[3]
[18202] clientloop.c:577 select(8,59)=1 41012 us W[3]
[18202] clientloop.c:577 select(8,59)=1 64691 us R[3]
[18202] clientloop.c:577 select(8,59)=1 42540 us W[3]
[18202] clientloop.c:577 select(8,59)=1 59321 us R[3]
[18202] clientloop.c:577 select(8,59)=1 32638 us W[3]
[18202] clientloop.c:577 select(8,59)=1 86610 us R[3]
[18202] clientloop.c:577 select(8,59)=1 25583 us W[3]
[18202] clientloop.c:577 select(8,59)=1 57985 us R[3]
[18202] clientloop.c:577 select(8,59)=1 33620 us W[3]
[18202] clientloop.c:577 select(8,59)=1 65790 us R[3]
[18202] clientloop.c:577 select(8,59)=1 47241 us W[3]
100MB                              3% 3584KB   3.5MB/s   00:26 ETA[
18202] clientloop.c:577 select(8,59)=1 72785 us R[3]
[18202] clientloop.c:577 select(8,59)=1 43191 us W[3]
[18202] clientloop.c:577 select(8,59)=1 58374 us R[3]
```

開発：[3]というのは、サーバと接続しているソケットのファイルディスクリプタです。
一方、高速な7.7の場合。

```
[18677] clientloop.c:557 select(7,59)=1 51259 us R[3]
[18677] clientloop.c:557 select(7,59)=1 10732 us R[3]
[18677] clientloop.c:557 select(7,59)=1 13437 us R[3]
100MB                              0%    0     0.0KB/s   --:-- ETA[
18677] clientloop.c:557 select(7,59)=1 10078 us W[3]
[18677] clientloop.c:557 select(7,59)=1 10849 us W[3]
[18677] clientloop.c:557 select(7,59)=1 11552 us W[3]
[18677] clientloop.c:557 select(7,59)=1 10066 us W[3]
[18677] clientloop.c:557 select(7,59)=1 13704 us W[3]
[18677] clientloop.c:557 select(7,59)=1 12787 us W[3]
[18677] clientloop.c:557 select(7,59)=1 10003 us W[3]
[18677] clientloop.c:557 select(7,59)=1 10073 us W[3]
[18677] clientloop.c:557 select(7,59)=1 10131 us W[3]
100MB                             16%   15MB  15.3MB/s   00:05 ETA[
18677] clientloop.c:557 select(7,59)=1 13581 us W[3]
[18677] clientloop.c:557 select(7,59)=1 17447 us W[3]
100MB                             30%   29MB  15.1MB/s   00:04 ETA[
18677] clientloop.c:557 select(7,59)=2 10901 us R[3]W[3]
100MB                             45%   43MB  15.0MB/s   00:03 ETA[
18677] clientloop.c:557 select(7,59)=1 11283 us W[3]
[18677] clientloop.c:557 select(7,59)=2 12001 us R[3]W[3]
[18677] clientloop.c:557 select(7,59)=1 12390 us W[3]
[18677] clientloop.c:557 select(7,59)=2 14756 us R[3]W[3]
[18677] clientloop.c:557 select(7,59)=2 12475 us R[3]W[3]
```

開発：7.8では10msを大きく超えることが無く、また入出力が同時にreadyになることが多い。一方7.7では待ちが50ms程度かかることが多く、しかも送受信一方ずつしかreadyにならない。これが10倍の性能差の原因かと思われます。

社長：監視しているファイルディスクリプタの数の 8 と 7 の違いは何でしょう？

開発：selectの第一引数の仕様はなんだか混乱していて、あまり真剣に設定したことが無いですね。実際このケースでは、待っているのは [3] だけなんです。

開発：えーと。ここは単にサーバと通信を双方向中継してるだけだと思うんですが、なんでこんな複雑そうに見えるかと言うと、sshはスレッドを使ってないんですね。PAMスレッドでしか使ってない。それで、I/Oをnonblockingにして、自力で双方向の中継を実装している。こういう実装法だと、上り下り両方がreadyの場合にも、片方ずつしか中継できないんで、nonblockingとは言え、並列性が活かしきれない気がします。特にマルチコアでは。

社長：なぜにそのような実装になっているんでしょうね？

開発：sshは1995年以前に作られたもののようです。当時は標準のスレッド技術が無かったと思います。それに、スレッドだとメモリを余計に食うというのはありますよね。あとは、やりたい場合にはですが、上り下りの間の同期を制御しやすいとか。

社長：確かに…　DeleGateでも色々なスレッドとお付き合いしました。

開発：えーと。もっと恐ろしいことがわかりました。scp から ssh にデータを送っているわけですが、これは pipe でできていました。

社長：pipe ってバッファサイズが 4KBとか…

開発：最近では64KBのようですが。でも、ファイルを読んでただ送っているだけなわけです。なら、SSHにファイルディスクリプタを継承すれば良いのにと思うのですが。まあ、パイプ経由でコマンドも送るんでしょうけど。

社長：Windowsではソケットのハンドルはファイルディスリプタ的に継承されないですしね。

＊　＊　＊

開発：ふぁあぁあ…　よく寝ました。

社長：それでどうなりました？

開発：sshのコードを3時間ばかり格闘したのですが、どうもその、サーバとの通信状況でパケットの送受信の戦略が変わるようでもあり、あるいはその先のTCPの転送経路が何か学習するのか、振る舞いがつかめませんでした。とにかく、送受信のバッファを大きくすれば良いというようなシンプルな話では内容です。

開発：ただ、動作原理はわからないですが、途中にアプリケーション層でのTCP中継器を挟めば問題が回避できることは確かです。

開発：そんなわけで、sshがサーバにコネクトした後に、そのソケットに一段フィルター的に中継プロセスをかませることにしました。

isfilter-1.0.c ダウンロード

isfilter-1.0.c ダウンロード

開発：特長は、select もスレッドも使ってないところです（笑）

開発：でもってこれを、sshconnect.c に挿し込みます。

```
diff -c ../*8p1/sshconnect.c sshconnect.c
*** ../openssh-7.8p1/sshconnect.c       2018-08-23 14:41:42.000000000 +0900
--- sshconnect.c        2020-08-05 20:47:38.158957487 +0900
***************
*** 577,582 ****
--- 577,584 ----
            sizeof(on)) < 0)
                error("setsockopt SO_KEEPALIVE: %.100s", strerror(errno));

+ #include "../isfilter.c" //@@2020-0805 SatoxITS
+
        /* Set the connection. */
        if (ssh_packet_set_connection(ssh, sock, sock) == NULL)
                return -1; /* ssh_packet_set_connection logs error */
```

開発：そして、環境変数 _ISFENV を定義して ssh や scp をやると、このフィルタが挿入されるという仕掛けです。やってみます。

```
MacMini% scp ~/100MB dg9:
100MB                                   100%   95MB   1.1MB/s   01:25
MacMini% _ISFENV= ./scp ~/100MB dg9:
setsockopt TCP_NODELAY: Operation not supported on socket
100MB                                   100%   95MB  12.3MB/s   00:07
```

基盤：おお、10倍速達成！

開発：socketpairにAF_UNIXを使っていると、macOSではNODELAYが出来ないと文句を言って来ますすが、まあそもそものTCP立ち上がりの問題なので、実害はありません。

開発：この対策は、レイテンシがめっちゃ長い外部サーバにはそれほど効きません。USサイトとはRTTが130ms程度ですが、まあ2倍速くらいかなという感じです。

```
MacMini% ping -c 3 us1
PING us1 (44.232.35.67): 56 data bytes
64 bytes from 44.232.35.67: icmp_seq=0 ttl=229 time=129.358 ms
64 bytes from 44.232.35.67: icmp_seq=1 ttl=229 time=129.729 ms
64 bytes from 44.232.35.67: icmp_seq=2 ttl=229 time=129.704 ms

--- us1 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 129.358/129.597/129.729/0.169 ms
MacMini%
MacMini% ./scp ~/20MB us1:
20MB                                    100%   19MB 779.4KB/s   00:25
MacMini% _ISFENV= ./scp ~/20MB us1:
setsockopt TCP_NODELAY: Operation not supported on socket
20MB                                    100%   19MB   1.8MB/s   00:10
```

開発：最果ての地、フランクフルトではこんな感じ。

```
MacMini% ping -c 3 de1
PING ns2 (52.59.53.30): 56 data bytes
64 bytes from 52.59.53.30: icmp_seq=0 ttl=233 time=253.012 ms
64 bytes from 52.59.53.30: icmp_seq=1 ttl=233 time=253.248 ms
64 bytes from 52.59.53.30: icmp_seq=2 ttl=233 time=253.332 ms

--- ns2 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 253.012/253.197/253.332/0.135 ms
MacMini% ./scp ~/20MB de1:
20MB                                  100%   19MB 512.7KB/s   00:38
MacMini% _ISFENV= ./scp ~/20MB de1:
setsockopt TCP_NODELAY: Operation not supported on socket
20MB                                  100%   19MB   1.4MB/s   00:14
MacMini%
```

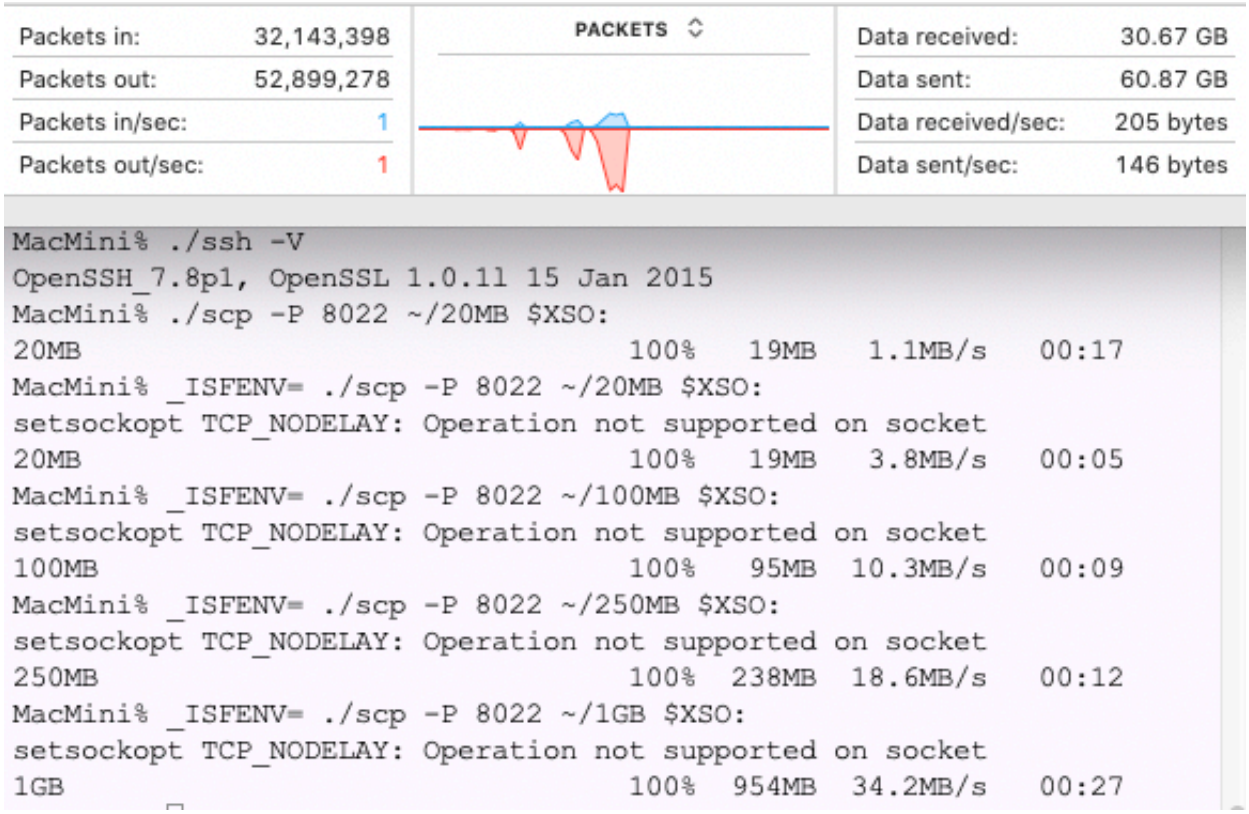
社長：まあでも、2倍高速化すれば十分じゃないですか。

基盤：遠隔でも、ライトセール間なら5MB/sくらい出ますので、一旦ライトセール東京に持っていって、そこからさらに先に配るのが良さそうですね。

開発：そういえば現在、xsoへのアップロードは30MB/sくらい出ます。ところが、macOSでやってみたら、3MB/sくらいしか出ないんです。

```
u18% ./ssh -V
OpenSSH_7.8p1, OpenSSL 1.0.2n  7 Dec 2017
u18% ./scp -P 8022 -i ~/.ssh/xso3.pem ~/1GB $XSO:
1GB                                    18%   74MB   1.1MB/s   04:47 ETA^
Cu18%_ISFENV= ./scp -P 8022 -i ~/.ssh/xso3.pem ~/1GB $XSO:
1GB                                   100%  394MB  28.0MB/s   00:14
```

開発：ところは腹いせにでっかいファイルを送ってやったら、ジョジョに加速して、34MB/s に到達したりするんです。



基盤：不思議過ぎる…　何がどうこでどう調整されているのやら。

社長：相手によってヒューリスティックに戦略を変える必要があるのかも知れないですね。

開発：奥が深そうです。

社長：それはそうと、せっかく作った isfilter プログラム、なんかの形で公開しましょう。

基盤：このブログじゃだめなんですか？w

— 2020-0805 SatoxITS