株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.07.17 06:31:37 +09'00'

2020年7月16日 投稿者: SATOXITS

クラウドVM間ディスク共有

基盤：さてそれでは、ライトセール上のVM間のディスク共有を行いたいと思います。

開発：性能的にどうですかね。

基盤：レスポンス的には、pingでのrttが、リージョン内では 0.5ms、遠いリージョン間では 150ms ～ 250ms といったところで、2桁以上の違いがあります。一方でスループット的には、ファイル転送がリージョン内では60MB/s、リージョン間では6MB/s といったところで、1桁程度の違いです。

開発：引っ越し時の大量転送は別として、6MB/s というのは昔のHDDと同等の性能ですから、まあ使い物にはなりますね。ですが、RTT 250ms だと、たとえば1ファイルを作るのに1秒とかかかったりしそうだし、作業用にランダムアクセスするのも厳しそうです。もっとも、RAMでバッファしてくれればそこそこ吸収できるのかも知れません。

基盤：で、リモートファイルアクセスのプロトコルですが。

開発：まずはNFSじゃないですかね。一番軽そうだし。設定も、nfsd 側で exportfs して、クライアント側で mount.nfs するだけだし。もともと公開しているファイルを共有するだけなので、セキュリティについては特に考慮する必要はありません。アクセス元をIPアドレスで制限すれば十分と思います。

基盤：では、まず nfsd 側を設定します…　/etc/exports…　空っぽですね。 / とだけ書けば良いのかな。で、/usr/sbin/rpc.nfsd…　ああ、なんか怒られてます。ではおおせにしたがって /proc/fs/nfsd をマウント。おもしろい考え方ですね。/usr/sbin/rpc.nfsd。netstat -a|grep LISTEN。sunrpc立ち上がった模様。で、mkdir /nfs/localhostして、mount.nfs localhost /nfs/localhost…　なんか怒られてますね。mount.nfs4 -v localhost:/ /nfs/localhost …。Connection refused。あれ？サーバプロセス数を指定しないといけないみたいですね。/usr/sbin/rpc.nfsd 4。netstat -a|grep LISTEN。ps ax|grep nfsd。nfsd 受付開始です。ふたたび sudo

mount.nfs4…　おっと、Permission denied が出ました。/etc/exports を / * にして exportfs -ra…　変わらないですね。/var/lib/nfs/etab を見る…

開発：ぶぶー。タイムアウトです。ぐぐりましょう。amazon linux nfs …。ああ、service nfs start で起動するみたいですよ。

基盤：sudo service nfs start。mount.nfs4…　繋がりました。

```
amalin% df
Filesystem       1K-blocks    Used Available Use% Mounted on
devtmpfs            493880      64    493816   1% /dev
tmpfs               504568       0    504568   0% /dev/shm
/dev/xvda1        41217324 1185816  39931260   3% /
localhost:/local  41217408 1185792  39931264   3% /nfs/localhost
```

基盤：1時間ばかりロスしてしましました。

社長：教訓としては、①昔のUnixと違ってサーバを管理するコマンドがある。②意地を張らずにググる。ですね。

基盤：では、共有用のフォルダを /pub とかなんとかにして。exports に加えて exportfs -ra。で、他のマシンから mount.nfs …。ああ、ファイアウォールを通さないといけないですね。111番 TCP？だめ。2049番 TCP？通りました。シドニー・ムンバイ間開通です。

基盤：東京ーシドニー間でも接続。mountしてファイルをtar xf … やはり1ファイル作成に1秒というところですね。いつもの　time openssl rand 100000000 > 100MB… スループット的には6MB/s 〜 10MB/s ですね。

社長：これTCPモードみたいですが、UDPでやったらどうなりますかね。あ、でもその前にお昼に行きますか。

＊　＊　＊

基盤：ふぁあ。よく寝た。さてまず、NFSのパケットを覗いてみましょう。tcpdump あれ、ないんですね。yum install tcpdump。あった。では。クライアント側で echo > x。

```
07:38:55.676309 IP in1.3385712612 > Me.nfs: 136 getattr fh 0,0/22
07:38:55.676394 IP Me.nfs > in1.3385712612: reply ok 196 getattr NON 2 ids 0/9 sz 0
07:38:55.815850 IP in1.820 > Me.nfs: Flags [.], ack 2673, win 848, options [nop,nop,
TS val 4002551730 ecr 1456924167], length 0
07:38:55.815896 IP in1.3402489828 > Me.nfs: 216 getattr fh 0,0/22
07:38:55.815973 IP Me.nfs > in1.3402489828: reply ok 300 getattr NON 5 ids 0/18 sz 0
07:38:55.955491 IP in1.3419267044 > Me.nfs: 176 getattr fh 0,0/22
07:38:55.956868 IP Me.nfs > in1.3419267044: reply ok 220 getattr NON 3 ids 0/34 sz 0
07:38:56.096473 IP in1.3436044260 > Me.nfs: 176 getattr fh 0,0/22
07:38:56.098282 IP Me.nfs > in1.3436044260: reply ok 132 getattr NON 3 ids 0/38 sz 0
07:38:56.238145 IP in1.3452821476 > Me.nfs: 160 getattr fh 0,0/22
07:38:56.238239 IP Me.nfs > in1.3452821476: reply ok 132 getattr NON 3 ids 0/9 sz 0
07:38:56.420605 IP in1.820 > Me.nfs: Flags [.], ack 3473, win 848, options [nop,nop,
TS val 4002552335 ecr 1456924729], length 0
```

開発：55.67 に最初のパケットが来て、56.42 のackで終わってますね。0.75秒。

基盤：UDPでつなぐにはどうすればいいんですかね…　man mount… udp というマウントオプションがあるような。…　そんなオプションは無いって言われますね。NFSv4ではTCPがデフォになったそうで、UDPをどう扱うかは実装次第なのかも知れません。

開発：うーん…　そうですか。まあ理想的に1往復で行ったとしても 250msの壁は超えられないわけですしね。結果を待たないという意味では async オプションが良いのかな？

基盤：async でマウント。echo > x。

```
08:52:34.156195 IP in1.1729861635 > Me.nfs: 136 getattr fh 0,0/22
08:52:34.156280 IP Me.nfs > in1.1729861635: reply ok 196 getattr NON 2 ids 0/9 sz 0
08:52:34.295664 IP in1.nmap > Me.nfs: Flags [.], ack 130929, win 848, options [nop,n
op,TS val 68368973 ecr 1461342756], length 0
08:52:34.296489 IP in1.1746638851 > Me.nfs: 216 getattr fh 0,0/22
08:52:34.296559 IP Me.nfs > in1.1746638851: reply ok 300 getattr NON 5 ids 0/18 sz 0
08:52:34.435932 IP in1.1763416067 > Me.nfs: 144 getattr fh 0,0/22
08:52:34.436012 IP Me.nfs > in1.1763416067: reply ok 68 getattr NON 2 ids 0/20 sz 0
08:52:34.575446 IP in1.1780193283 > Me.nfs: 180 getattr fh 0,0/22
08:52:34.576583 IP Me.nfs > in1.1780193283: reply ok 220 getattr NON 3 ids 0/34 sz 0
08:52:34.716064 IP in1.1796970499 > Me.nfs: 160 getattr fh 0,0/22
08:52:34.716159 IP Me.nfs > in1.1796970499: reply ok 132 getattr NON 3 ids 0/9 sz 0
08:52:34.896693 IP in1.nmap > Me.nfs: Flags [.], ack 131665, win 848, options [nop,n
op,TS val 68369575 ecr 1461343316], length 0
```

基盤：変わらないですね。

開発：まあ、クライアント側が待たないだけですからね。連続上書きしたり、書いて速攻同じものを読んだ時にどうなるかとか。

社長：因果は高速を超えられませんからね。RTT 250ms の世界で頑張っても仕方がないかな。ローカルならHDDでも10ms以下ですからね。

開発：並列化とかパイプラインをやらないとダメそうですね。上のレイヤでもっと処理をまとめて通信往復回数を少なくするか… バッファリングかキャッシュか。だいぶ昔にiSCSI というのが流行ったと思うんですが、あれは一体どうなってたんでしょうね？？

基盤：しかしこれだけ通信が遅いと、暗号化処理がネックになることはなさそうですね。大きなファイルの転送のスループットで言えば、scp でも cp/NFSでも同速度。

```
India% time scp $XXX -p 100MB au1:
100MB                         100%   95MB   6.7MB/s   00:14

real    0m16.234s
user    0m0.552s
sys     0m0.149s
India% time cp -p 100MB /nfs/au1

real    0m16.283s
user    0m0.000s
sys     0m0.070s
India%
```

基盤：大きなファイルの上書きではやはりrsyncですね。

```
India% time rsync 100MB au1:/pub/100MB

real    0m12.897s
user    0m0.861s
sys     0m0.248s
India% time rsync 100MB au1:/pub/100MB

real    0m3.313s
user    0m0.522s
sys     0m0.020s
```

社長：sshは、まあコンフィギュレーション次第なのかも知れませんが、ネゴシエーションのための行き来が多すぎる。リモートに高速に貼ったり切ったりするものでは無いですよね。

開発：NFSもTCP上でって言うなら、SSLで通せば簡単ですね。

社長：まあそういうことで、100msを超えるRTTの世界というのを久しぶりに見た気分です。使い方的には、ローカルなVM間ではアクティブなファイルの共有に、リモートなVM間ではアーカイブかバックアップ置き場程度に、という感じでしょうか。

基盤：では、当社世界５拠点間での/pubなデータの共有を設定しておきます。

＊　＊　＊

基盤：ということで、作業終了しました。まず東京本部からの視点。

```
jp1$ df
Filesystem           1K-blocks     Used Available Use% Mounted on
devtmpfs                493880       64    493816   1% /dev
tmpfs                   504568        0    504568   0% /dev/shm
/dev/xvda1            41217324 26800512  14316564  66% /
jp1:/uni.map/zzz/pub  41217408 26800512  14316672  66% /uni.map/jp1/pub
us1:/uni.map/zzz/pub  41217408 29138176  11979008  71% /uni.map/us1/pub
de1:/uni.map/zzz/pub  41217408 28978304  12138880  71% /uni.map/de1/pub
au1:/uni.map/zzz/pub  41217408  1649792  39467392   5% /uni.map/au1/pub
in1:/uni.map/zzz/pub  41217408  1381632  39735552   4% /uni.map/in1/pub
jp1$
```

基盤：オレゴン支部から。

```
us1% df
Filesystem           1K-blocks     Used Available Use% Mounted on
devtmpfs                493880       64    493816   1% /dev
tmpfs                   504568        0    504568   0% /dev/shm
/dev/xvda1            41217324 29138168  11978908  71% /
jp1:/uni.map/zzz/pub  41217408 26800768  14316416  66% /uni.map/jp1/pub
us1:/uni.map/zzz/pub  41217408 29138176  11979008  71% /uni.map/us1/pub
de1:/uni.map/zzz/pub  41217408 28978304  12138880  71% /uni.map/de1/pub
au1:/uni.map/zzz/pub  41217408  1649792  39467392   5% /uni.map/au1/pub
in1:/uni.map/zzz/pub  41217408  1381632  39735552   4% /uni.map/in1/pub
us1%
```

基盤：フランクフルト支部から。

```
de1$ df
Filesystem           1K-blocks     Used Available Use% Mounted on
devtmpfs                493880       64    493816   1% /dev
tmpfs                   504568        0    504568   0% /dev/shm
/dev/xvda1            41217324 28978272  12138804  71% /
jp1:/uni.map/zzz/pub  41217408 26800512  14316672  66% /uni.map/jp1/pub
us1:/uni.map/zzz/pub  41217408 29138176  11979008  71% /uni.map/us1/pub
de1:/uni.map/zzz/pub  41217408 28978304  12138880  71% /uni.map/de1/pub
au1:/uni.map/zzz/pub  41217408  1649792  39467392   5% /uni.map/au1/pub
in1:/uni.map/zzz/pub  41217408  1381632  39735552   4% /uni.map/in1/pub
de1$
```

基盤：以下略。ネーミングは暫定です。

開発：これでいちいちscpとかしなくて良くなって、仕事がやりやすくなりますね。

社長：あとは、ミツツボアリをどう作るかということですね。

— 2020-0716 SatoxITS