

フリーのDNSストレスツールの紹介

dnsperf (開発元 Nominum),
dnstcpbench(開発元 Nether Labs)

2013/7/18

SCSK株式会社

服部 成浩 (s.hattori@scsk.jp)

自己紹介

- ・ Nominum社の商用DNS,DHCPソフトウェアの技術を担当しています
 - Nominumの回し者ではありません
-
- プライベート
 - DNS(Nominum除く)
 - unbound, PowerDNS, BIND10とたわむれています
 - DNS以外
 - KVM, Linux Container, OpenStack, Python を学習中

DNSテストツールのを2つ紹介

- 権威サーバ、キャッシュサーバのサイジングや負荷をかけている時の挙動確認に役立つツールを2つ紹介
 - dnsperf (開発元 Nominum)
 - dnstcpbench (開発元 Nether Labs)

Nominum DNSPerf

- DNSPerfとは
 - Nominum社が開発したオープンソースのDNSストレスツール
- dnsperfダウンロード先
 - www.nominum.com → Support → Measurement Tools
 - ソースコード
 - RPMパッケージ (RHEL4,5,6 32bit)

64bit OS(CentOS6 64bit)上で、DNSperfをビルドする方法

- 64bit版はRPMパッケージがないので、ソースコードからビルドする必要がある
 - ソースコードからビルドができない、という質問を受けることがあるのCentOS6.4 64bit版でのビルド方法をご紹介します
- 32bit版はRPMパッケージがあるので32bit版はインストールが簡単

DNSperf ビルド(CentOS 6.4 64bit)

- ビルド前の準備 (その1~3)

- その1

- dnsperfのビルドに必要なRPMをyumでインストール

```
# yum install -y bind bind-devel bind-libs libcap-devel  
libxml2-devel gnuplot gcc make
```

DNSperf ビルド(CentOS 6.4 64bit)

- その2
 - hmacsha.h をBINDのソースコードから取得。/usr/include/isc ディレクトリにコピー

```
# wget
```

```
ftp://ftp.isc.org/isc/bind9/9.9.3-P1/bind-9.9.3-P1.tar.gz
```

```
# tar xzvf bind-9.9.3-P1.tar.gz
```

```
# cp bind-9.9.3-P1/lib/isc/include/isc/hmacsha.h  
/usr/include/isc
```

64bit OS(CentOS6 64bit)上で、DNSperfをビルドする方法

- その3
- シンボリックリンクの作成

```
# ln -s /lib64/libgssapi_krb5.so.2.2  
/usr/lib64/libgssapi_krb5.so
```

```
# ln -s /usr/lib64/libcrypto.so.10 /usr/lib64/libcrypto.so
```


64bit OS(CentOS6 64bit)上で、DNSperfをビルドする方法

- やっところ、dnsperfをビルド

```
# tar xzvf dnsperf-src-2.0.0.0-1.tar.gz
# cd dnsperf-src-2.0.0.0-1
# sh configure
# make
# make install
```

```
# /usr/bin/dnsperf -h
DNS Performance Testing Tool
Nominum Version 2.0.0.0
```

Fedora 19 だとyum install dnsperf でインストール可能

DNSPerfの基本的な使い方

- 基本的な使い方
 - `# dnsperf -s 192.168.0.10 -d query_list.txt`

Statistics:

```
Queries sent:      33740
Queries completed: 33740 (100.00%)
Queries lost:      0 (0.00%)
```

```
Response codes:   NOERROR 33740 (100.00%)
Average packet size: request 32, response 112
Run time (s):     0.642633
Queries per second: 52502.750403
```

```
Average Latency (s): 0.001300 (min 0.000295, max 0.012613)
Latency StdDev (s): 0.000831
```

- `-d` クエリリストファイル
- `-s` ターゲットのIPアドレス
- クエリリストファイルの書式

```
# cat query_list.txt
www1.foo.bar A
www2.foo.bar A
```

DNSPerf2.0からマルチスレッドに対応

- 1クライアントで2スレッド使用
 - 送信で1スレッド、受信で1スレッド
 - 2クライアントだと4スレッド
- 指定方法: -c クライアント数
 - クライアント数 2 `./dnssperf -c 2 -d query.txt -s IP`
 - クライアント数 100 `./dnssperf -c 100 -d query.txt -s IP`
- デフォルト(-c を指定しない場合) は クライアント数は1

DNSPerf2.0からマルチスレッドに対応

- 1クライアントの定義
 - 送信元UDPポート番号が異なる
 - 注意: 送信元IPアドレスは同じ
- -c 1 のときは、Transaction ID(65536)数が、同時に名前解決中クエリ(outstanding queries)の上限値となる
- c オプションで名前解決中のクエリを65536以上模擬できる
 - c 2 : $65536 * 2 =$ 約13万

送信しているQPSをリアルタイムで表示(dnsperf)

- -S 表示間隔(秒)
 - テスト中にdnsperf が送信しているQPSを、“-S 数値”で指定した間隔(秒)で表示
- # dnsperf -s 192.168.10.12 -d query_list.txt -S 1 -c 4
- DNS Performance Testing Tool
- Nominum Version 2.0.0.0
- [Status] Command line: dnsperf -s 192.168.10.12 -d query_list.txt -S 1 -c 4
- [Status] Sending queries (to 192.168.10.12)
- [Status] Started at: Tue Jul 16 16:37:49 2013
- [Status] Stopping after 1 run through file
- 1373960270.261348: 67693.546824 ←エポック時間: QPS
- 1373960271.262496: 67608.385573

指定したQPSで負荷をかける(dnsperf)

- -Q ターゲットQPS
 - # dnsperf -Q 1000
 - 一定のQPSを処理しているときに設定変更などのオペレーションを実施してもサービス断がないかなどの確認に便利
- # dnsperf -s 192.168.10.12 -d query_list.txt -c 4 -Q 50000 -S 1
- [Status] Stopping after 1 run through file
- 1373969191.797204: 49966.308558 ← だいたい50,000qpsを維持
- 1373969192.797533: 49991.552779
- 1373969193.798084: 50005.446999

dnstcpbench

- Power DNSの開発元 Nether Labs が開発
- マルチスレッドに対応している
- TCP fallback(UDP→TCP)に対応している
 - 先ほど紹介した dnsperf は TCP fallbackに対応していません

dnstcpbenchビルド方法

Ubuntu12.04 64bit

- dnstcpbenchをビルドするために必要なパッケージをインストール
 - # apt-get install autoconf automake bison flex g++ libboost-all-dev libtool make pkg-config ragel zlib1g-dev git

dnstcpbenchビルド方法

Ubuntu12.04 64bit

```
# git clone https://github.com/PowerDNS/pdns.git
# cd pdns/
# ./bootstrap
# ./configure --with-modules="" --without-lua
# cd pdns/ ( ← pdsn/pdns ディレクトリへ移動)
# make dnstcpbench
```

“make dnstcpbench”を実行したディレクトリにdnstcpbench が生成される

Fedora 19 だと、yum install pdns-tools でインストール可能

dnstcpbench オプション

- --udp-firstオプションなし (デフォルト)
 - 最初からTCPで問い合わせる
- --udp-firstオプションあり

まずUDPで問い合わせる

- レスポンスがTruncatedしていない場合
 - 名前解決終了
- レスポンスがTruncatedしている場合
 - TCPで問い合わせる (TCPフォールバック)

dnstcpbench

OSのチューニング

- オープンできるファイルディスクリプタ数を増やす
 - # ulimit -n 65536
- ファイルディスクリプタが枯渇すると、下記のエラーがでる
 - Network error: Too many open files

dnstcpbench使用例

- 使用例

```
# ./dnstcpbench --udp-first --file=query_list.txt 192.168.0.100 53
```

- --udp-first
 - 最初はUDPで問い合わせ
 - --file=クエリリストファイル
 - 最後に、IPアドレス ポート番号

- クエリリストファイルの書式

FQDN クエリタイプ

```
# cat query_list.txt  
www.foo.bar A  
www.foo.bar A
```

dnstcpbench

テスト結果表示例

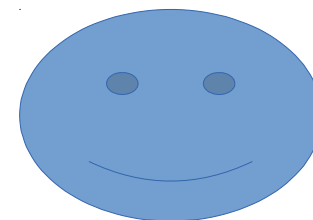
- UDPクエリUDP → TCP fallback となったクエリは、UDP → TCP fallbackのセットで1クエリとカウントされる
 - # ./dnstcpbench --udp-first --file=query_list.txt 192.168.0.100 53
 - Average qps: 943.396, median qps: 1008.31
 - Average UDP latency: 1190.06usec, median: 1106.41usec
 - Average TCP latency: 62321.8usec, median: 2659.29usec
 - OK: 48229, network errors: 0, other errors: 0
 - Timeouts: 1771
 - **Truncateds: 50000**, auth answers: 48229

dnstcpbench UDPクエリのキャプチャ抜粋

- eDNS0オプションはつけないみたい
 - オン、オフのオプションもない
- DOビットもつかないみたい
 - オン、オフのオプションもない
- --udp-first を指定したときのUDPクエリ
 - Domain Name System (query)
 - Transaction ID: 0x0000
 - Flags: 0x0000 (Standard query)
 - 0... .. = Response: Message is a query
 - .000 0... .. = Opcode: Standard query (0)
 -0. = Truncated: Message is not truncated
 -0 = Recursion desired: Don't do query recursively
 -0. = Z: reserved (0)
 -0 = Non-authenticated data OK: Non-authenticated data is unacceptable
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - www.foo.bar: type A, class IN
 - Name: www.foo.bar
 - Type: A (Host address)
 - Class: IN (0x0001)

dnstcpbench まとめ

- UDP(512bytes) → TCP fallback の試験が可能
 - eDNS0は未対応
- eDNS0, DNSSECにも対応してくれるさらにうれしい



まとめ: DNSストレスツール比較

queryperf vs. dnsperf vs. dnstcpbench

- dnsperf を使うのならマルチスレッド対応のdnsperf 2.0がおすすめ
- ツールによって使用できる機能に差があるので、試験内容に応じツールの使い分けが必要

	queryperf (BIND9.9.3-P1ソースコード のqueryperf)	dnsperf 2.0	dnstcpbench (バージョン情報なし。2013/6月末頃の dnstcpbench)
マルチスレッド対応	X	○	○
Dynamic Update	X	○ (TSIGも可)	X
TCP fallback	X	X	○ UDP(512) → TCP
eDNS0	○	○	○
DO bit	○	○	○
指定したQPSの負荷	○	○	X