

## 技術ノート KGTN 2011030201

### 現象

GGNetworkCheck とはどんなツールなのか？

### 説明

GG のサーバ側とクライアント側で動作させるプログラムで、クライアント側主導で TCP/IP のヘルスチェック（接続性確認）を行います。クライアント側は所定の秒数で PING メッセージを送信し（ログに記録も残しません）、サーバ側は PING メッセージを受信するとログに記録するとともに、応答メッセージをクライアント側に送信します。クライアント側は応答メッセージを受信すると、その情報をログに記録します。所定秒数内に相手からのメッセージが届かない、または TCP/IP レベルでコネクションが切断された場合は、エラーとしてログに記録されます。この2つのログ（クライアント側とサーバ側）を付き合わせることで、ネットワークの状況を確認することが出来ます。

GGNetworkCheck は、GMSERVICE との併用で自動監視が出来るように、コンソールアプリケーションとして作成されています。GGNetworkCheck は、1つのプログラムの中にサーバ機能とクライアント機能の両方を持っており、逆方向のヘルスチェックも簡単に行うことが出来ます。さらに、サーバ機能は「多」クライアントからの同時アクセスも可能な実装になっています。

### 設定ファイル

設定ファイル GGNetworkCheck.ini の default セクションで、以下のパラメタを設定して下さい。

設定項目	説明
mode	S = サーバとして起動 / C = クライアントとして起動
host	サーバのIPアドレス (IPv4 / IPv6)
port	サーバの待受けポートの番号
interval	ヘルスチェックの間隔 (秒)
repeat	ヘルスチェックの回数 (注: -1は無限回)
sleep	コネクション切断後の再チェックまでの間隔 (秒)
clTimeout	クライアント時の送受信のタイムアウト時間 (秒)
svTimeout	サーバ時の送受信タイムアウト時間 (秒) 注: clTimeout +5秒以上を指定すること
logd	ログを保存するフォルダ (絶対パス)
prefix	送信メッセージの先頭に付加する文字列
clTimeAdj	ログに記録されるクライアント時間の微調整 (±3600秒まで)
svTimeAdj	ログに記録されるサーバ時間の微調整 (±3600秒まで)

設定項目は、下記のコマンドの引数で上書きすることが出来ます。

設定項目	説明
-s	サーバとして起動
-c	クライアントとして起動
-h address	サーバのIPアドレス (IPv4/IPv6)
-hp port	サーバの待受けポートの番号
-i sec	コネクション切断後の再チェックまでの間隔 (秒)
-r times	ヘルスチェックの回数 (注: -1は無限回)
-l dir	ログを保存するフォルダ (絶対パス)
-x msg	送信メッセージの先頭に付加する文字列

同じ設定ファイル GGNetworkCheck.ini を使用し、サーバ側では

```
C:\¥GraphOn¥Tools> GGNetworkCheck -s
```

で起動し、クライアント側では

```
C:\¥GraphOn¥Tools> GGNetworkCheck -c
```

で起動することが出来ます。この方法を行うことで、サーバ側とクライアント側の設定が矛盾する等の問題を避けることが出来ます。特に必要がない限り、同じ設定ファイルを使用するようにして下さい。

## ログファイル

クライアント側のログファイルの読み方は、以下の通りです。

### 1. 正常な場合

```
[2011-03-05 00:00:17] Test#1 CL@00:00:17 ->  
[2011-03-05 00:00:17] Test#1 CL@00:00:17 -> SV@00:00:20 -> CL@00:00:17 (46ms)
```

クライアントが 00:00:17 にヘルスチェックのパケットを送信し、サーバが 00:00:20 (サーバ側の時計での時刻です) に受信し、その応答がクライアントへ 00:00:17 に届いたことを意味します。応答までに要した時間は 46 ミリ秒です。

### 2. 遅延した場合

```
[2011-03-05 00:01:22] Test#1 CL@00:01:22 ->  
[2011-03-05 00:01:27] Test#1 CL@00:01:22 -> SV@00:01:30 -> CL@00:01:27 (4766ms)
```

クライアントが 00:01:22 にヘルスチェックのパケットを送信し、サーバが 00:01:30 (サーバ側の時計での時刻です) に受信し、その応答がクライアントへ 00:01:27 に届いたことを意味します。応答までに要した時間は 4766 ミリ秒です。

### 3. 障害が発生した場合

```
[2011-03-05 00:04:21] Test#1 CL@00:04:21 ->  
[2011-03-05 00:04:32] ERROR: recv('192.168.102.198', '8491') -> Failed
```

クライアントが 00:04:21 にヘルスチェックのパケットを送信したものの、それに対する応答がサーバから届かなかったことを意味します。サーバ側にこのパケットが届いているかどうかは、サーバ側のログで確認することが出来ます。可能性としては、1) クライアントは送信したがサーバに届いていない、2) サーバ側に届いて応答も送信したがクライアントに届いていない、の2つのパターンがあります。

#### 4. その他

[2011-03-05 00:01:15] \*CLIENT\* Timeout = 10sec

[2011-03-05 00:01:15] INFO: socket('192.168.102.198', '8491') -> Connected

サーバと接続出来た場合は, 上記ログが記録されます.

[2011-03-06 01:05:21] ERROR: connect('192.168.102.198', '8491') -> Failed

サーバと接続出来なかった場合は, 上記ログが記録されます.

Last reviewed: Mar 09, 2011

Status: DRAFT

Ref: NONE

Copyright © 2011 kitASP Corporation