# Ensuring PingER is Working

J. Zurawski – July 2013

## Symptom:

You visit the PingER GUI, and note that some of the addresses (or all of the addresses) are in the Non-Active area. E.g.:



Figure - PingER with Non-Active Test

## Solution:

The problem is in how PingER resolves DNS/Host names. Before each round of tests it will do a DNS lookup. If for some reason DNS fails (or is slow), it will rely on IP addresses. If DNS works, it will rely on hostnames. This means that the database may start to build up records for one, the other, or both. In the case of ‘both’ this will have an impact on the display GUIs, which will inevitably pick the wrong one to display, and thus show things as not working.

The solution involves manually getting into the database, finding the suspect entries, and then deleting them. After deleting them, revisit the same web page and things will look correct immediately:



Figure - PingER After DB Steps

## Solution Steps:

1. Log on to host via SSH as root, or sudo to root.
2. Open Mysql



Figure - Mysql

1. Type **use pingerMA;** to get the appropriate database.
2. You can run another command to get a listing of all tables: **show tables;**
3. The first step is to see what is in the host table. Type **select \* from host;** The output may look like this:

**mysql> select \* from host;**

**+------+-------------------------+---------------+---------+**

**| host | ip\_name | ip\_number | ip\_type |**

**+------+-------------------------+---------------+---------+**

**| 1 | 164.113.32.33 | 164.113.32.33 | ipv4 |**

**| 2 | ps-wsu-lt.kanren.net | 164.113.32.33 | ipv4 |**

**| 3 | ps-bryant-lt.kanren.net | 164.113.32.9 | ipv4 |**

**| 4 | ps-ku-lt.kanren.net | 164.113.32.17 | ipv4 |**

**| 5 | ps-ksu-lt.kanren.net | 164.113.32.25 | ipv4 |**

**+------+-------------------------+---------------+---------+**

**5 rows in set (0.00 sec)**

1. In the above example, we are going to test the hypothesis that host #1 (the one without a DNS resolution) is causing a conflict with host #2. Another thing to note is that we are physically logged into this host, so if we have data with conflicting ‘src’ hostnames, we will have display problems. To test this further, we need to see what is in the metaData table using this command: **select \* from metaData;**

**mysql> select \* from metaData;**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**| metaID | src\_host | dst\_host | transport | packetSize | count | packetInterval | ttl |**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**| 1 | 2 | 4 | icmp | 1000 | 10 | 1 | 255 |**

**| 2 | 2 | 3 | icmp | 1000 | 10 | 1 | 255 |**

**| 3 | 2 | 5 | icmp | 1000 | 10 | 1 | 255 |**

**| 4 | 2 | 2 | icmp | 1000 | 10 | 1 | 255 |**

**| 5 | 1 | 5 | icmp | 1000 | 10 | 1 | 255 |**

**| 6 | 1 | 4 | icmp | 1000 | 10 | 1 | 255 |**

**| 7 | 1 | 3 | icmp | 1000 | 10 | 1 | 255 |**

**| 8 | 1 | 2 | icmp | 1000 | 10 | 1 | 255 |**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**8 rows in set (0.00 sec)**

1. Our hypothesis is correct, we have conflicting source host records. Another hypothesis we are making is that src\_host 1 (e.g. metaID 5, 6, 7, and 8) is the problem. We can test that with a quick look at the most recent months worth of data using this command: **select count(\*) from data\_201307 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**mysql> select count(\*) from data\_201307 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**+----------+**

**| count(\*) |**

**+----------+**

**| 0 |**

**+----------+**

**1 row in set (0.03 sec)**

1. The previous result tells us that no data was collected for this set of tests. We can contrast this with another command to test the opposite of the hypothesis: **select count(\*) from data\_201307 where metaID=1 or metaID=2 or metaID=3 or metaID=4;**

**mysql> select count(\*) from data\_201307 where metaID=1 or metaID=2 or metaID=3 or metaID=4;**

**+----------+**

**| count(\*) |**

**+----------+**

**| 50933 |**

**+----------+**

**1 row in set (0.03 sec)**

1. The previous result tells us that data was collected for this set. This means that the first set (e.g. metaIDs 5, 6, 7 and 8 and host 1) need to be removed from all database tables. Once we do that, data should be displayed properly. We do this in a couple of steps:
	1. Find out the names of all data tables
	2. Delete the offending data IDs from the tables
	3. Delete the offending data IDs from the metaData table
	4. Delete the offending host from the host table.
2. To find out all of the data tables, do this command: **show tables;**

**mysql> show tables;**

**+--------------------+**

**| Tables\_in\_pingerMA |**

**+--------------------+**

**| data |**

**| data\_201202 |**

**| data\_201203 |**

**| data\_201204 |**

**| data\_201205 |**

**| data\_201206 |**

**| data\_201207 |**

**| data\_201208 |**

**| data\_201209 |**

**| data\_201210 |**

**| data\_201211 |**

**| data\_201212 |**

**| data\_201301 |**

**| data\_201302 |**

**| data\_201303 |**

**| data\_201304 |**

**| data\_201305 |**

**| data\_201306 |**

**| data\_201307 |**

**| host |**

**| host\_tmp |**

**| metaData |**

**| metaData\_tmp |**

**+--------------------+**

**23 rows in set (0.00 sec)**

1. We need to delete the metaIDs from each of these tables, the following commands will do that:

**delete from data\_201307 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**delete from data\_201306 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**...**

**delete from data\_201202 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

Some commands will show this:

**mysql> delete from data\_201304 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**Query OK, 17 rows affected (0.13 sec)**

Others will show this (if there was no data to delete):

**mysql> delete from data\_201304 where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**Query OK, 0 rows affected (0.11 sec)**

It is important to do this from all tables.

1. The next step is to remove records of the IDs from the metaData table. We know that the host id is ‘1’, so we can remove all of the places where we see that as the ‘src\_id’. First read the table:

**mysql> select \* from metaData;**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**| metaID | src\_host | dst\_host | transport | packetSize | count | packetInterval | ttl |**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**| 1 | 2 | 4 | icmp | 1000 | 10 | 1 | 255 |**

**| 2 | 2 | 3 | icmp | 1000 | 10 | 1 | 255 |**

**| 3 | 2 | 5 | icmp | 1000 | 10 | 1 | 255 |**

**| 4 | 2 | 2 | icmp | 1000 | 10 | 1 | 255 |**

**| 5 | 1 | 5 | icmp | 1000 | 10 | 1 | 255 |**

**| 6 | 1 | 4 | icmp | 1000 | 10 | 1 | 255 |**

**| 7 | 1 | 3 | icmp | 1000 | 10 | 1 | 255 |**

**| 8 | 1 | 2 | icmp | 1000 | 10 | 1 | 255 |**

**+--------+----------+----------+-----------+------------+-------+----------------+------+**

**8 rows in set (0.00 sec)**

Then delete where the src\_host is 1:

**mysql> delete from metaData where src\_host=1;**

**Query OK, 4 rows affected (0.04 sec)**

A similar command would have also worked:

**mysql> delete from metaData where metaID=8 or metaID=7 or metaID=6 or metaID=5;**

**Query OK, 0 rows affected (0.11 sec)**

1. Finally we need to get rid of the host from the host table. Read the table:

**mysql> select \* from host;**

**+------+-------------------------+---------------+---------+**

**| host | ip\_name | ip\_number | ip\_type |**

**+------+-------------------------+---------------+---------+**

**| 1 | 164.113.32.33 | 164.113.32.33 | ipv4 |**

**| 2 | ps-wsu-lt.kanren.net | 164.113.32.33 | ipv4 |**

**| 3 | ps-bryant-lt.kanren.net | 164.113.32.9 | ipv4 |**

**| 4 | ps-ku-lt.kanren.net | 164.113.32.17 | ipv4 |**

**| 5 | ps-ksu-lt.kanren.net | 164.113.32.25 | ipv4 |**

**+------+-------------------------+---------------+---------+**

**5 rows in set (0.00 sec)**

Then we want to delete the host with the id of 1:

**mysql> delete from host where host=1;**

**Query OK, 1 row affected (0.01 sec)**

1. After you follow these steps, you should not have to restart any services. Visit the web page again and see if the previously ‘dead’ data is available:



Figure - After doing DB Steps