

O'REILLY®
OSCON
open source convention



MySQL Through SNMP

Gerardo "Gerry" Narvaja
@seattlegaucho

Agenda

- Introduction to our MySQL SNMP agent (sort of) fork
 - Why SNMP
 - Why MySQL agent
 - Why *pass_persist*
 - Why fork
- Short introduction on how to measure "bytes behind master"
 - Just a bunch of formulas
- Review / Interpret real life charts
 - Batch process
 - Slave load
 - Partially Stopped Slave

Why #1

- SNMP is a widely adopted standard for monitoring
 - <http://en.wikipedia.org/wiki/Snmp>
 - Widely supported: OpenNMS, Nagios, Cactii, OpenView, Unicenter, etc.
 - `pass_persist`: see *man 5 snmp.conf*
- MySQL SNMP agent
 - Google the term or ...
 - <http://www.masterzen.fr/software-contributions/mysql-snmp-monitor-my>
 - Uses Net-SNMP subagent

Why #2

- More than just a branch: fork
 - Use *pass_persist* instead of Net-SNMP
 - Support for older RH5 / CentOS packages
 - Bug fixes
 - Add our own functionality: BBM
 - Transformed the original agent into a module

Code Snippet #1

- `stderr` confuses the *pass_persist* protocol ... get rid of it

```
...
    openlog('mysql-ppsnmp', LOG_PID | LOG_PERROR, LOG_DAEMON);
    # close off STDERR if not connected to a tty (being run interactively)
    unless (-t STDERR) {
        open(STDERR, ">/dev/null");
    }
...

```

Code Snippet #2

- Non-blocking input from **stdin**

```
sub pp_read_command {
    my (@ready,@ret);
    @ready = $select->can_read(0.25);
    return unless scalar(@ready) > 0;
    ...
    }
    next if ($ret[0] =~ m/^PING/);
    if (index($ret[0], 'get') == 0) {
        # ensure we have the command and an oid number
        while (scalar @ret < 2) {
            $fh->blocking(1);
            push @ret, $fh->getline;
            $fh->blocking(0);
        }
        next;
    }
    ...
}

return \@ret;
}
```

Code Snippet #3

- output to **stdout**

```
sub pp_handle_cmd {
  my $buf = shift;
  while (my $line = shift(@$buf)) {
    chomp($line);
    if ($line eq 'PING') {
      $stdout->print("PONG\n");
      next;
    }
    elsif ($line eq 'set') {
      # slurp off the remainder of the set command
      my ($set_oid,$set_val) = splice(@$buf,0,2);
      # tell the agent we cannot 'set' anything
      $stdout->print("not-writable\n");
      next;
    }
    elsif ($line eq 'get') {
      my ($get_oid) = shift(@$buf);
      chomp($get_oid);
      $stdout->print(pp_set_value($get_oid));
    }
  }
  ...
}
```

BBM – Bytes Behind Master

- Seconds Behind Master only tells half the story
 - Collection of timestamps on the binary logs
 - Doesn't reflect the data volume
- Bytes Behind Master
 - SHOW SLAVE STATUS doesn't show all
 - Need to get information from the master
- On the master
 - SHOW BINARY LOGS

BBM – Slave Status

- Show Slave Status:

Master_Log_File: mysql-bin.003078
Read_Master_Log_Pos: 898475023

Last position read
from the master

Relay_Master_Log_File: mysql-bin.003078
Exec_Master_Log_Pos: 898475023

Position in the
master for last
write on the slave

BBM – Master Binary Logs

- Show Binary (Master) Logs:

```
+-----+-----+
| Log_name          | File_size  |
+-----+-----+
| mysql-bin.003071  | 1073742251 |
| mysql-bin.003072  | 1073742259 |
| mysql-bin.003073  | 776376749  |
| mysql-bin.003074  | 1073741982 |
| mysql-bin.003075  | 1110910261 |
| mysql-bin.003076  | 1073742812 |
| mysql-bin.003077  | 1073742944 |
| mysql-bin.003078 | 900425131 |
+-----+-----+
```

BBM – The Math

- If `Relay_Master_Log_File == Master_Log_File`
 - `Read_Master_Log_Pos – Exec_Master_Log_Pos`
- If `Relay_Master_Log_File < Master_Log_File`
 - Get binary log file list from master
 - $BBM1 = \text{Size of (Relay_Master_Log_File)} - \text{Exec_Master_Log_Pos}$
 - If `Relay_Master_Log_File` is more than 1 file behind
 - $BBM2 = \text{Sum (other log files' size)}$
 - $BBM3 = \text{Read_Master_Log_Pos}$
 - $BBM = BBM1 + BBM2 + BBM3$

Code Snippet #4

- The simple case

```
...  
# log file the master is still writing to  
if (defined($result->{'master_log_file'}) and $result->{'master_log_file'} ne "" and  
    $result->{'master_log_file'} eq $result->{'relay_master_log_file'}) {  
    $status{'slave_lag_bytes'} = ($result->{'read_master_log_pos'} - $result->{'exec_master_log_pos'});  
    ...  
}
```

Code Snippet #5

- The non-simple case

...

```
} elsif (defined($result->{'master_host'}) and defined($result->{'master_port'})) {
    my $mdsn;
    if ($datasource =~ m/mysql_read_default_file/) {
        $mdsn = $datasource . ';host=' . $result->{'master_host'} . ';port=' . $result->{'master_port'};
    } else {
        $mdsn = 'DBI:mysql:host=' . $result->{'master_host'} . ';port=' . $result->{'master_port'};
    }
    # otherwise, try and connect to the master host with the
    # same credentials to add up how far ahead the master is
    my $mdbh = DBI->connect($mdsn, $dbuser, $dbpass, {PrintError => 0, PrintWarn => 0});
    if (defined($mdbh)) {
        my $mbinlogs = $mdbh->selectall_arrayref('SHOW BINARY LOGS');
        my (%binlogs,$offset);
        foreach my $row (@$mbinlogs) {
            $binlogs{$row->[0]} = $row->[1];
        }
        $offset = ($binlogs{$result->{'relay_master_log_file'}} - $result->{'exec_master_log_pos'});
        for my $binlog (sort keys %binlogs) {
            # skip any log files <= slave's current read file
            next unless (($binlog cmp $result->{'relay_master_log_file'}) == 1);
            # add up the byte totals for anything newer
            $offset += $binlogs{$binlog};
        }
        $status{'slave_lag_bytes'} = $offset;
    }
}
```

...

Other Files

- OpenNMS files:
 - MIB file: Contains the definition of each value being collected / monitored
 - .xml files (/opt/opennms/etc)
 - snmp-graph.properties
 - datacollection-config.xml
- You'll need to adapt them to your SNMP monitoring package

Code Snippet #6

- The configuration line in `/etc/snmp/snmpd.conf`

...

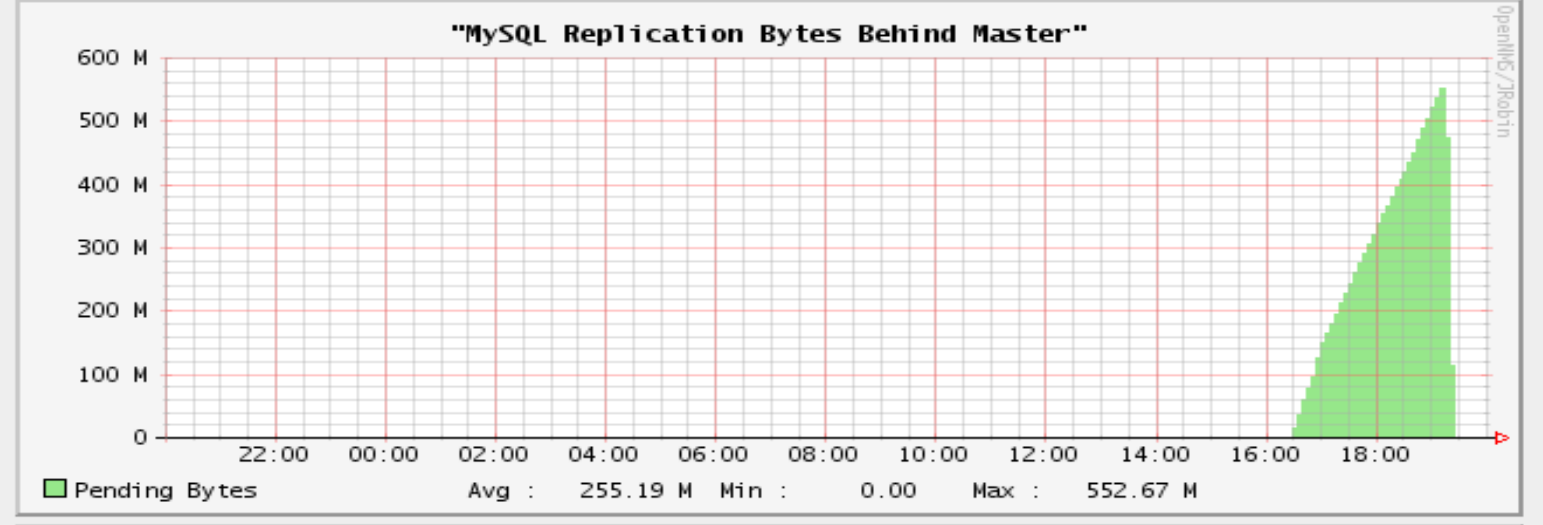
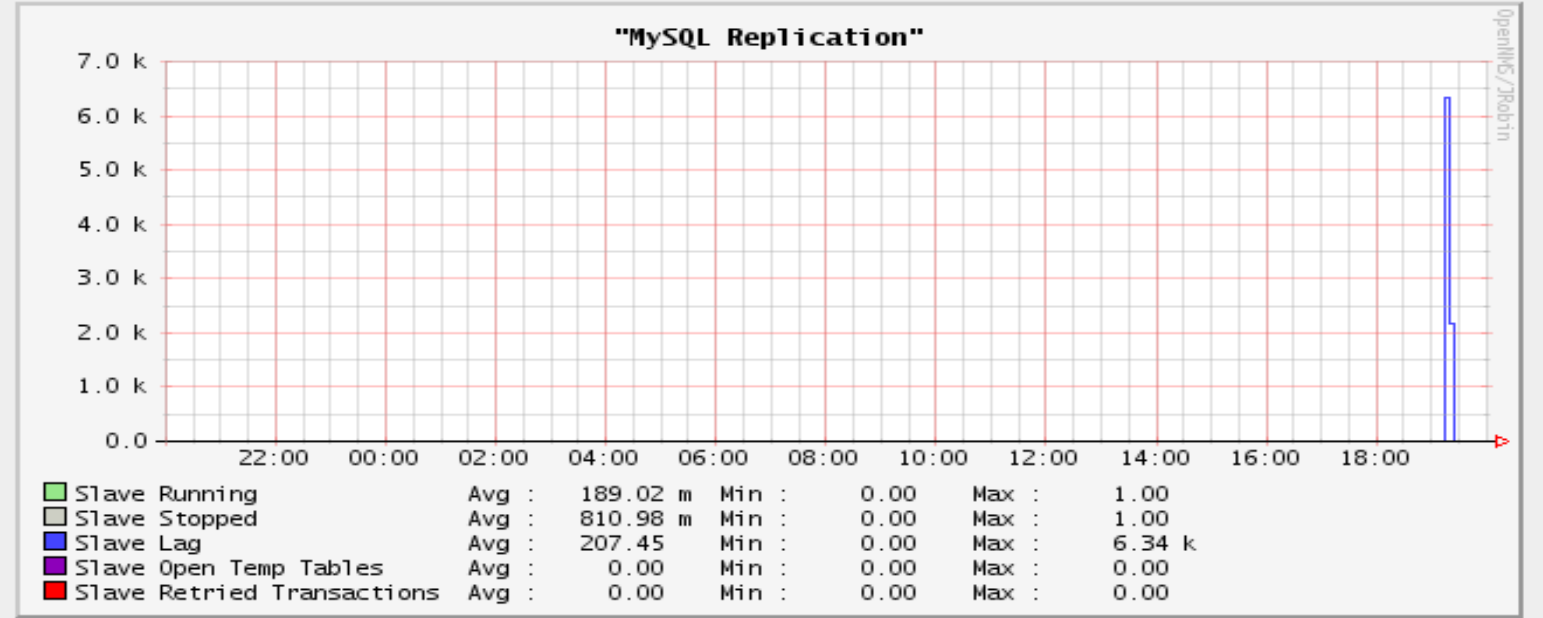
```
# Monitor the local MySQL instance
```

```
pass_persist .1.3.6.1.4.1.20267.200.1 /usr/share/snmp/mysql-ppsnmp \
```

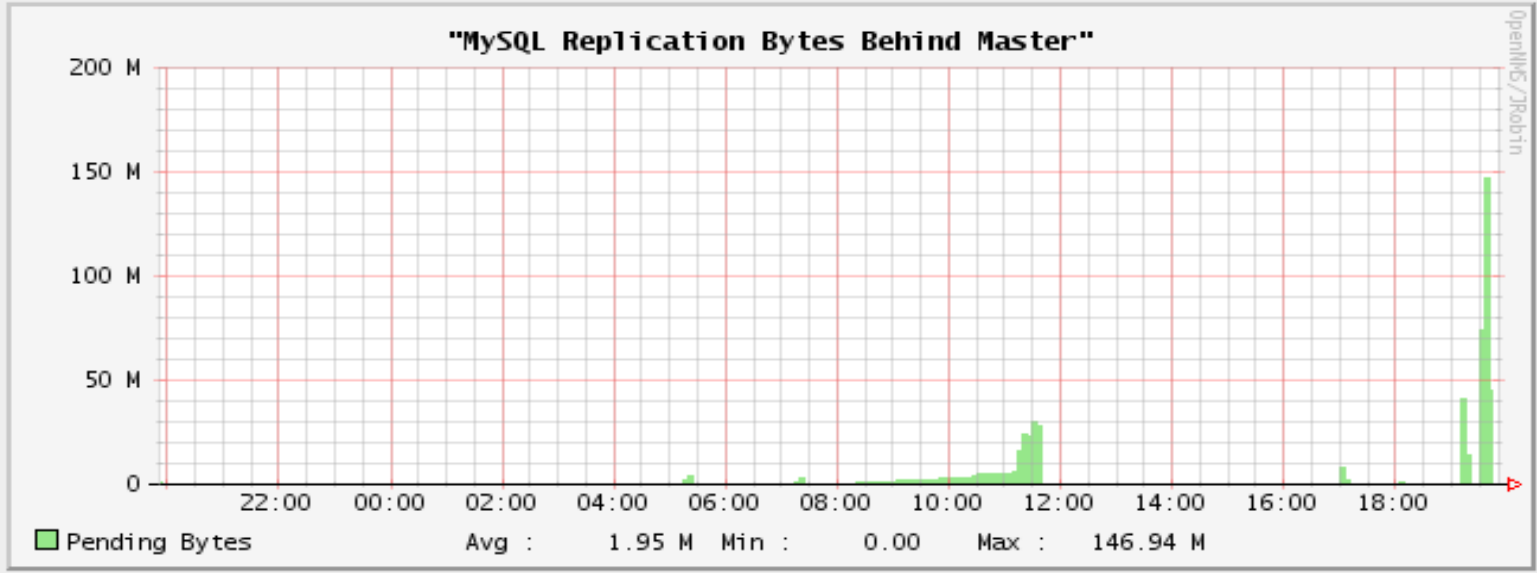
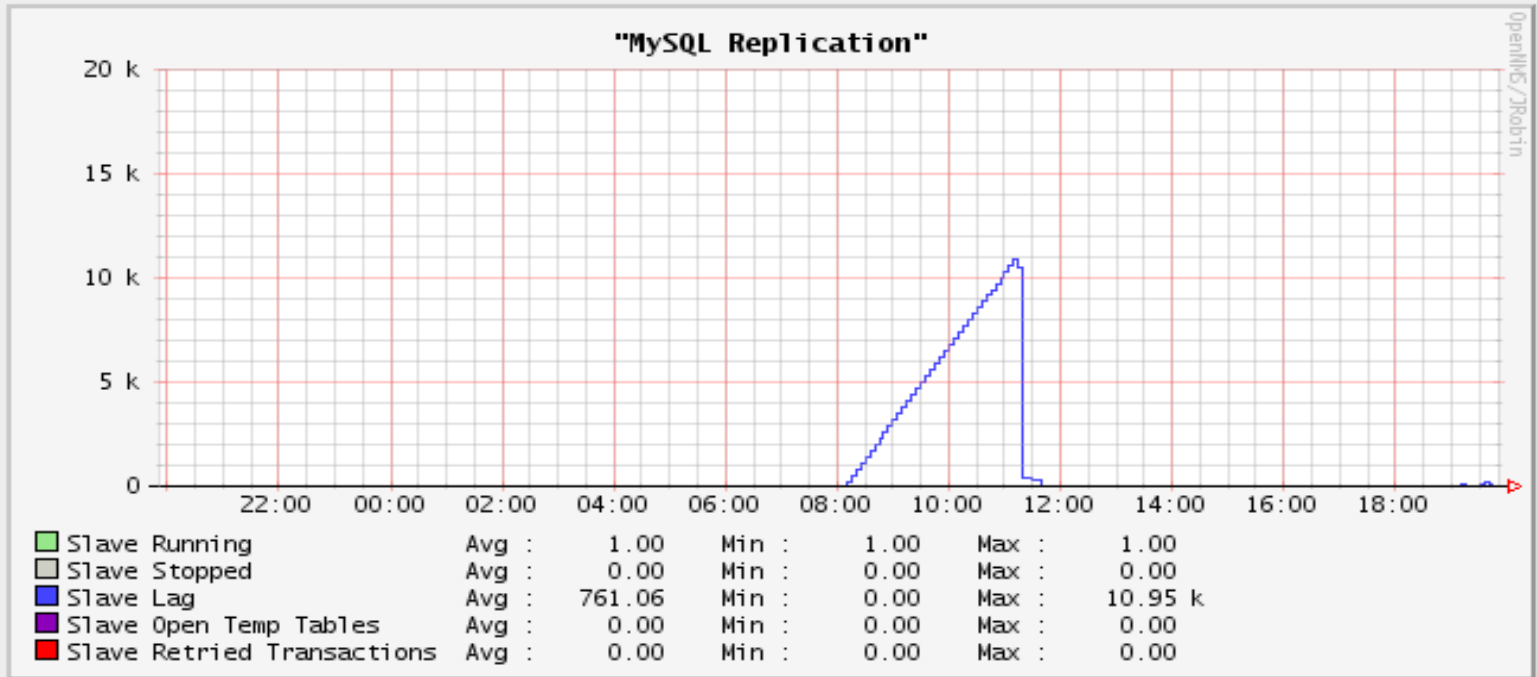
```
-c /opt/openmarket/mysql_utils/etc/snmp-my.cnf -m -s -h 127.0.0.1 -i -r 60 --procs --bbm
```

...

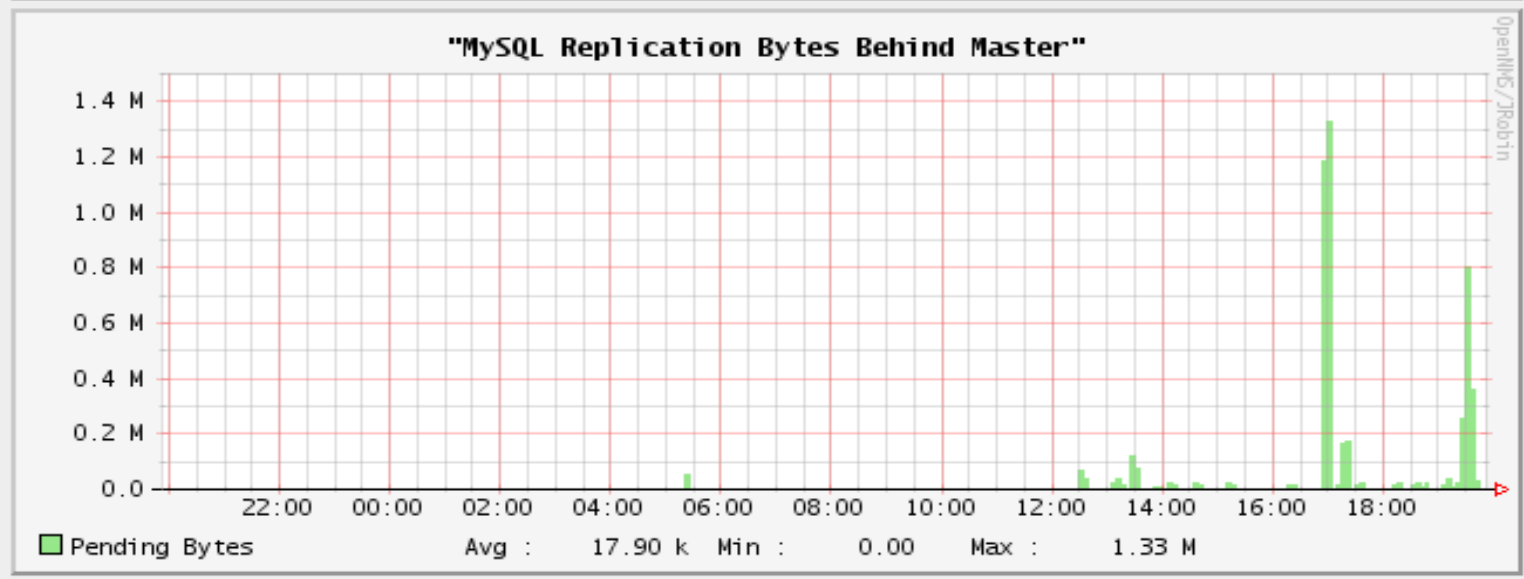
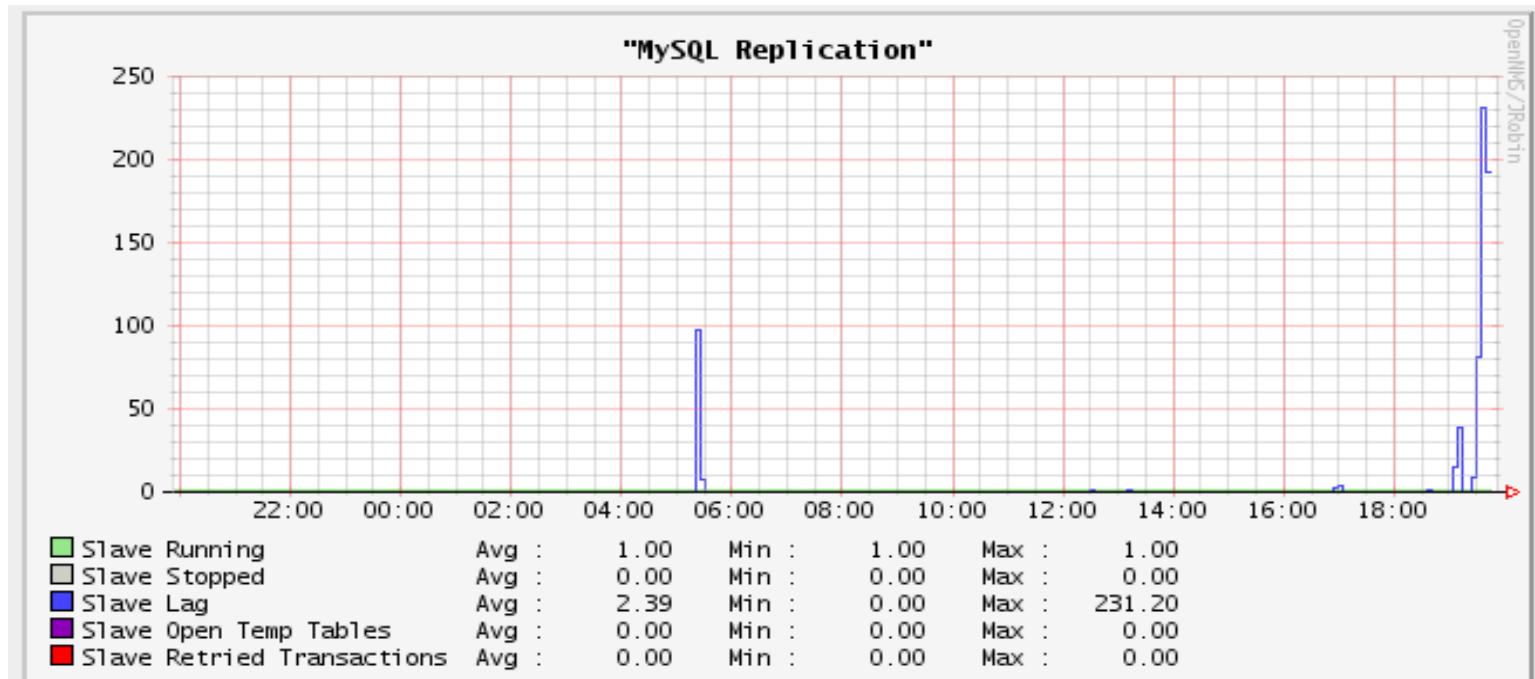
Enslaving a Server



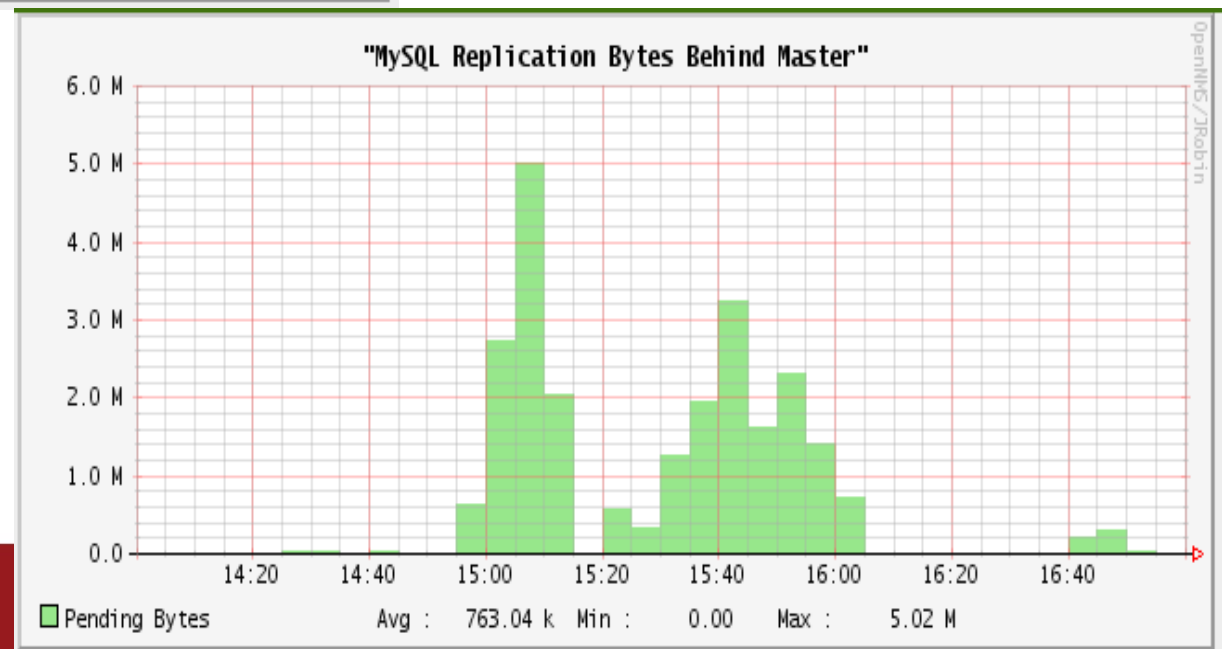
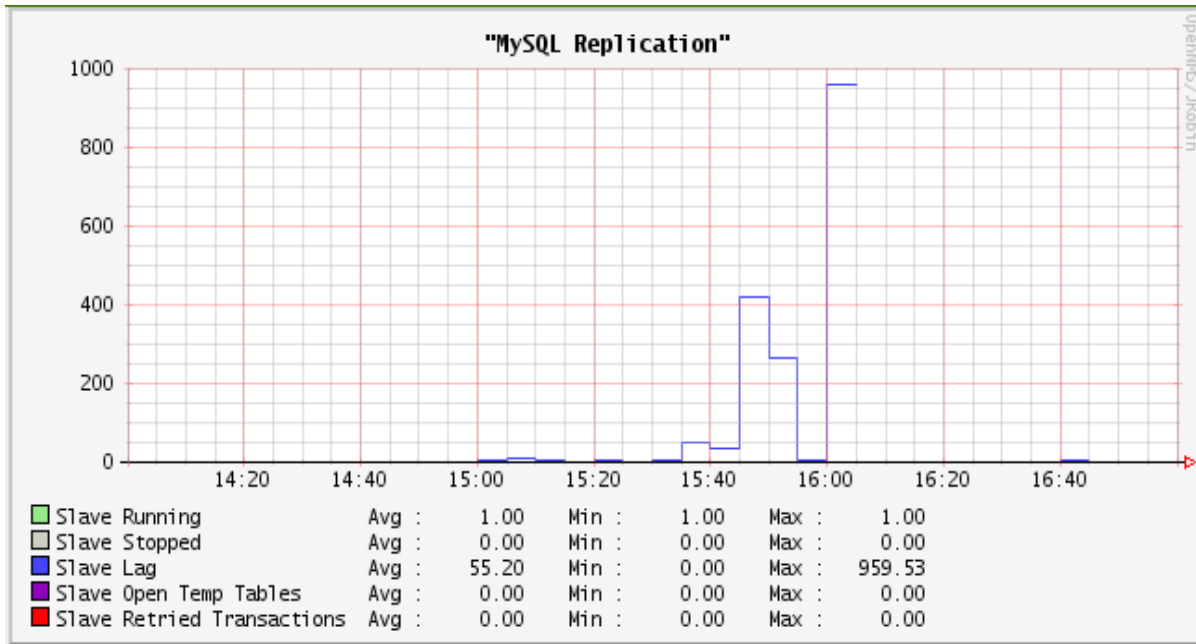
Nightly Backup



Just Another Batch



Trouble in Paradise?



Thank you very much ...

- Twitter: @seattlegaucho
- Blog: <http://mmatемate.blogspot.com>