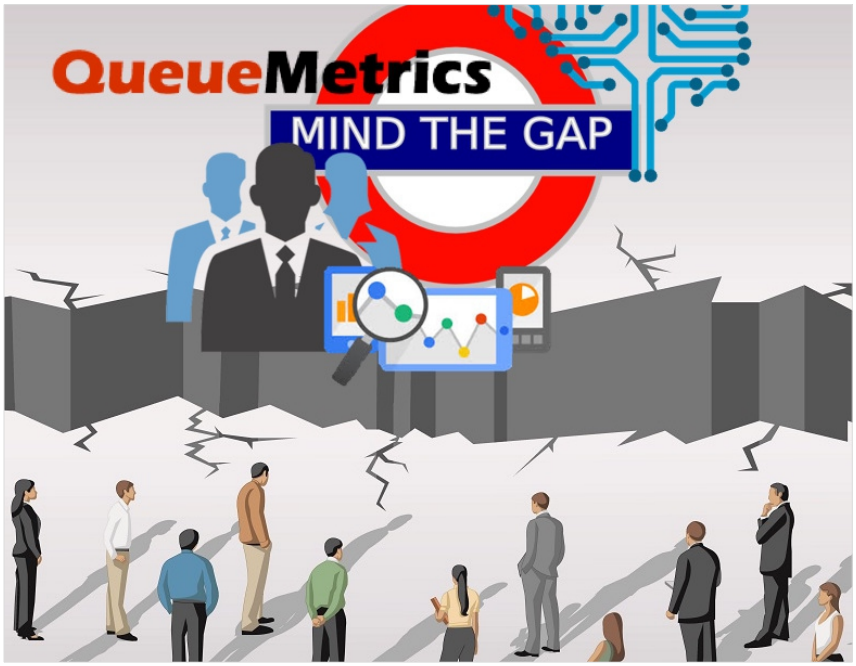
Loway presents

Advanced Troubleshooting: Filling a data gap in QueueMetrics



Missing Data

Are you missing data in QueueMetrics for a certain number of days? Do you have a “hole” in your data? In this tutorial we will take a look at how to import missing data in QueueMetrics, through the uniloader service.

QueueMetrics

QueueMetrics is a highly scalable monitoring and reporting suite that addresses the needs of thousands of contact centers worldwide and offers a broad range of integrated benefits like agent productivity monitoring, target measurement, conversion rates tracking, realtime campaign statistics analysis and an easy to use interface. It’s available on premise or as a cloud hosted solution service.

Import old data when a data gap is present

This procedure will show you how to import old queue_log data when a gap is present in the historical data.

This procedure is suggested when you have a few days of missing data, but the uniloader cannot upload it manually, because it has already uploaded more recent data than the missing data.

Keep in mind that the uniloader cannot upload data to a partition, if that partition already contains data that is more recent than the data you wish to import.

This is a safety measure enacted by the Uniloader, to avoid data corruption, but it can be a bother if data is missing. In this guide we will show you how to circumvent this, by operating on the database directly.

It is extremely important that you make a database backup before proceeding (see Database Backup section below), and that you don't attempt this procedure during working hours, as it could create some downtime, depending on the size of your QueueMetrics database.

Stop uniloader

It's very important that you stop the uniloader from running in the meantime.

```
service uniloader stop
```

Database backup

Make a copy of the current database, with the following instruction. Change 20190809 with the current date: YYYYMMDD

```
mysqldump -uqueuemetrics -pjavadude queuemetrics > qmBackup-20190809.sql
```

This should create a file, named qmBackup-20190809.sql. Make sure that the file exists, and that no errors were shown in the terminal, before proceeding.

Identify the gap

The first thing we need to do, is to identify the data gap.

Let's say, for example, that we are missing data from the 3rd of August until the 6th of August, and that data from the 7th of August onwards has been uploaded correctly.

Let's pick an arbitrary date and time that falls in the middle of the gap (Middle Point):

```
5th of August 2019 00:00 GMT + 2
```

and let's convert it into the corresponding linux timestamp. Timestamp Converter

```
1564956000
```

Now that we have a middle point, we need to find the last record that was uploaded before the gap starts (Point A), and the first record that was uploaded after the gap (Point B).

Basically, if the situation looks like this:

DATA DATA - Point A - GAP GAP - Middle Point - GAP GAP - Point B - DATA DATA

We will have the information on there the gap starts and ends.

To find out where point A (timestamp) is located, we need to access the MySql CLI

```
mysql -uqueuemetrics -pjavadude queuemetrics
```

NOTE: this is the default QueueMetrics username and password for MySQL, you might have changed that and replaced it with your own, in which case, please use your own credentials.

Identify the beginning of the gap (Point A)

```
select max(time_id) from queue_log where `partition` = 'PARTITION' AND time_id < MIDDLE POINT;
```

So in our case

```
select max(time_id) from queue_log where `partition` = 'P001' AND time_id < 1564956000;
```

in our case the result is

```
1564783200 (point A)
```

Identify the end of the gap (Point B)

```
select min(time_id) from queue_log where `partition` = 'PARTITION' AND time_id > MIDDLE POINT;
```

in our case

```
select min(time_id) from queue_log where `partition` = 'P001' AND time_id > 1564956000;
```

In our case the result is

```
1565128799 (point B)
```

Upload the missing data

We need to choose a new database partition to upload the missing data to, so that Uniloader won't have any issues uploading the data.

NOTE: the partition must be a new partition, with no previous data uploaded to.

We will call this partition : TEMP

Exit the MySQL CLI by typing and executing

```
exit
```

and run these operation in the terminal.

NOTE: execute these operations one at a time, not all together. Keep in mind that the process will not stop on it's own, so after the uniloader message with the number of inserted lines stop for a few seconds, interrupt the process manually with CTRL + C.

To make sure we get all the data, we upload data from the day before the gap starts, to the day after the gap ends.

NOTE: Make sure that you upload the queue_log files from the correct days, by choosing the correct filenames when setting the –src parameters.

NOTE: Remember to substitute QM's IP address in the commands, where IPADDRESS is shown.

```
uniloader --src /var/log/asterisk/queue_log-20190802 upload \
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"

uniloader --src /var/log/asterisk/queue_log-20190803 upload \
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"

uniloader --src /var/log/asterisk/queue_log-20190804 upload \
```

```
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"

uniloader --src /var/log/asterisk/queue_log-20190805 upload \
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"

uniloader --src /var/log/asterisk/queue_log-20190806 upload \
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"

uniloader --src /var/log/asterisk/queue_log-20190807 upload \
--uri "mysql:tcp(IPADDRESS:3306)/queuemetrics?allowOldPasswords=1" \
--login queuemetrics --pass javadude --token "TEMP"
```

Moving data from the TEMP partition to P001

Since our active partition is P001 (the one where the gap is), we need to move the newly uploaded data from TEMP to P001, being careful not to duplicate data in the process. To make sure we do everything correctly, we will only move data that is more recent than the beginning of the

gap, and older than the end of the gap. This should fill the data gap perfectly.

Go back to the MySQL CLI

```
mysql -uqueuemetrics -pjavadude queuemetrics
```

and run the following, substituting the correct values.

```
update queue_log set `partition` = 'ACTIVE_PARTITION'
where `partition` = 'TEMPORARY_PARTITION'
AND time_id > POINT_A AND time_id < POINT_B;
```

So, in our case,

```
update queue_log set `partition` = 'P001'
where `partition` = 'TEMP'
AND time_id > 1564783200 AND time_id < 1565128799;
```

This operation might take a while, depending on how many records you need to move from TEMP to P001.

Restart Uniloader

Once the operation is complete, we need to restart Uniloader, to keep on uploading the current data.

First, exit the MySQL CLI, then run this command:

```
service uniloader start
```

Now everything should be restored as intended.

QueueMetrics References

For more technical information about QueueMetrics call center solution please refer to the User Manual.

Visit www.queuemetrics.com for a 30 days full featured trial.

Attend our Free Webinars for a live demonstration of QueueMetrics.