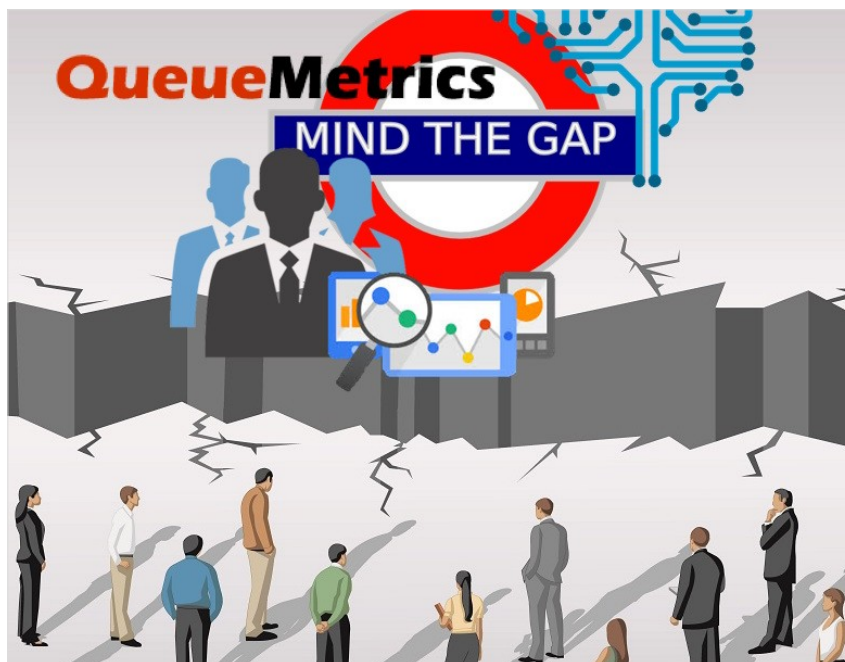


Расширенные возможности устранения неполадок: Заполнение пробелов в данных QueueMetrics



Пропущенные данные

Не хватает данных в QueueMetrics за определенное количество дней? Есть «дыра» в данных? В данном руководстве рассмотрим, как импортировать недостающие данные в QueueMetrics через сервис unloader.

QueueMetrics

QueueMetrics – это масштабируемый набор инструментов для мониторинга и отчетности, который удовлетворяет потребности тысяч контакт-центров по всему миру, и предлагает широкий спектр интегрированных функций, таких, как мониторинг производительности агентов, контроль целевых показателей, отслеживание конверсий,

анализ статистики кампании в реальном времени, и простой в использовании интерфейс. Он доступен как на отдельном сервере, так и в качестве облачного сервиса.

Импортировать старые данные, когда имеются пробелы в текущих данных

Данная процедура показывает, как импортировать старые данные из `queue_log`, когда существуют пробелы в исторических данных.

Данная процедура рекомендуется, когда есть пропущенные данные за несколько дней, но `unloader` не может выгрузить их вручную, потому что он уже выгрузил более свежие данные, чем пропущенные.

Имейте в виду, что `unloader` не может выгружать данные в раздел, если этот раздел уже содержит более новые данные, чем те, которые вы хотите выгрузить.

Это является мерой безопасности, принятой в `Unloader`, чтобы избежать повреждения данных, но это может доставлять неудобства, если данные отсутствуют. В данном руководстве показано, как это обойти, работая с базой данных напрямую.

Чрезвычайно важно сделать резервную копию базы данных перед этой процедурой (см. раздел Резервное копирование базы данных, расположенный ниже), и не пытаться выполнить эту процедуру в рабочее время, так как это может привести к некоторому простоя, зависящему от размеров базы данных.

Остановите unloader

Очень важно остановить `unloader` во время выполнения процедуры.

```
service unloader stop
```

Резервная копия базы данных

Сделайте копию текущей базы данных по следующей инструкции (измените 20190809 на текущую дату в формате: YYYYMMDD)

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

В результате должен быть создан файл с наименованием qmBackup-20190809.sql. Прежде чем продолжить, убедитесь, что файл существует, и что не было никаких ошибок.

Идентифицируйте пропуски

Первое, что нужно сделать, это идентифицировать пропуски в данных.

Допустим, для примера, что не хватает данных с 3 по 6 августа, и что данные, начиная с 7 августа, выгружались корректно.

Давайте выберем произвольную дату и время, попадающую в середину пропущенного периода (Средняя точка (Middle Point)):

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

и преобразуем ее в соответствующую временную метку linux (linux timestamp). [Timestamp Converter](#)

```
1564956000
```

Сейчас, когда есть средняя точка (middle point), нужно найти последнюю запись, которая была выгружена до начала разрыва (Point A), и первую запись, которая была выгружена после окончания разрыва (Point B).

В основном, если ситуация похожа на эту:

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

У нас будет информация, где разрыв начинается и заканчивается.

Чтобы найти point A (временная метка), нужен доступ к MySql CLI

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

ПРИМЕЧАНИЕ: это имя пользователя и пароль QueueMetrics для MySQL по умолчанию, но если они были изменены и заменены своими собственными, то пожалуйста, используйте свои учетные данные.

Определите начало разрыва (Point A)

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

Так, в данном случае

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

в результате получили

```
1564783200 (point A)
```

Определите конец разрыва (Point B)

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

в данном случае

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

в результате получили

```
1565128799 (point B)
```

Выгрузка пропущенных данных

Необходимо выбрать новый раздел базы данных для выгрузки пропущенных данных, чтобы у Uniloadер не было проблем с выгрузкой данных.

ПРИМЕЧАНИЕ: раздел должен быть новым разделом, в который не выгружались предыдущие данные.

Назовем данный раздел: TEMP

Выйдите из MySQL CLI, набрав и выполнив

```
exit
```

и запустите эти операции в терминале.

ПРИМЕЧАНИЕ: выполняйте эти операции по одной, а не все вместе. Имейте в виду, что процесс не остановится сам по себе, поэтому когда сообщения от uniloadер с числом вставленных строк остановится на несколько секунд, прервите процесс вручную, нажав CTRL + C.

Чтобы убедиться, что получим все данные, мы выгрузим данные, начиная за день до разрыва, и закончив днем позже окончания разрыва.

ПРИМЕЧАНИЕ: убедитесь, что выгрузили файлы queue_log за нужные дни, выбрав правильные имена файлов при установке параметров—src.

ПРИМЕЧАНИЕ: не забудьте подставить IP-адрес QM в командах вместо текста IPADDRESS.

```
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 queumetrics --pass javadude --token "TEMP"
```

```
unloader --src /var/log/asterisk/queue_log-20190807 upload \
```

```
--uri "mysql:tcp(IPADDRESS:3306)/queumetrics?  
allowOldPasswords=1" \
```

```
--login queumetrics --pass javadude --token "TEMP"
```

Переместите данные из раздела TEMP в P001

Так как активным разделом является P001 (тот, где есть пропущенные данные), необходимо переместить вновь выгруженные данные из TEMP в P001, стараясь не дублировать данные в процессе. Чтобы убедиться, что все делается правильно, будем перемещать только данные, более поздние, чем начало разрыва, и не позднее, чем конец разрыва. Таким образом пробел в данных должен быть заполнен.

Вернитесь назад в MySQL CLI

```
mysql -uqueumetrics -pjavadude queumetrics
```

и выполните следующую команду, подставляя нужные значения.

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

Так, в данном случае

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

Данная операция может занять некоторое время, в зависимости от того, сколько записей надо переместить из TEMP в P001.

Перезапуск Unloader

После завершения операции необходимо перезапустить Unloader для продолжения выгрузки текущих данных.

Вначале выйдите из MySQL CLI, затем выполните команду:

```
service unloader start
```

Теперь все должно быть восстановлено, как и задумывалось.

Ссылки

Более подробная техническая информация о решении для колл-центров QueueMetrics находится в [User Manual](#).

Посетите www.queuemetrics.com для получения 30-дневной бесплатной триальной версии.

Обратите внимание на [Free Webinars](#) с живой демонстрацией QueueMetrics.