## SRE 4 (03.12.2021) | Проблема с БД | Часть 3: SLI freshness

Вычисление freshness SLI происходит по универсальной формуле:

кол-во хороших событий / общее кол-во событий

Проблема в том, что при работе с replication delay у нас нет событий, таких как, например, запрос на сервер. Более того, метрика, которую мы создали на предыдущем шаге ( pg\_replication\_delay\_behind\_primary\_seconds ), имеет тип Gauge . Единственное, с чем мы можем работать, это со снятием показания как с событием. Или в терминологи Prometheus - scrape .

Определим "хорошее событие" как - Prometheus произвел измерение (scrape) и значение replication delay ниже 1 секунд. Заведем для это ряд метрик.

```
kind: PrometheusRule
apiVersion: monitoring.coreos.com/v1
metadata:
    name: prometheus-pgsql-slo-rules
    namespace: $city
spec:
    groups:
        - name: pgsql-slo-rules
        rules:
        - record: $city:job:pgsql_replica_replication_delay_is_below_threshold
        expr: pg_replication_delay_behind_primary_seconds{namespace="$city"} <= bool 1
        - record: $city:job:pgsql_replica_replication_delay_is_above_threshold
        expr: pg_replication_delay_behind_primary_seconds{namespace="$city"} > bool 1
```

- pgsql\_replica\_replication\_delay\_is\_below\_threshold эта метрика равна 1, когда replication delay меньше или равен пороговому значению
- pgsql\_replica\_replication\_delay\_is\_above\_threshold эта метрика равна 1, когда replication delay больше порогового значения

Можно использовать метрику в качестве порогового значения. Как это сделать рассказано в статье Using time series as alert thresholds.

Далее, просумировав pgsql\_replica\_replication\_delay\_is\_below\_threshold на промежутке времени (например 5 минут), мы получил количество раз, когда Prometheus измерил replication delay и он был в норме:

```
sum_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m])
```

Это наши "хорошие события".

Далее перейдем к "общему количеству событий".

Тут проще. Функция count\_over\_time позволяет измерить общее количество измерений (scrapes).

```
count_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m])
```

Далее нам остается только разделить эти два значения, чтобы получить нужную величину SLI:

```
100
  * sum_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m])
  / count_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m])
```

Таким образом мы получим SLI для каждой реплики отдельно. Чтобы получить общую статистику достаточно просуммировать значения.

```
100
  * sum(sum_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m]))
```

```
/ sum(count_over_time($city:job:pgsql_replica_replication_delay_is_below_threshold[5m]))
```

Это и есть наше SLI.

## Алерты

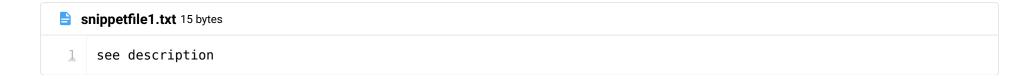
Значение pgsql\_replica\_replication\_delay\_is\_above\_threshold хорошо подходит для алертов.

Не забываем про \$city

```
kind: PrometheusRule
apiVersion: monitoring.coreos.com/v1
metadata:
 name: prometheus-pgsql-slo-alerts
 namespace: $city
spec:
 groups:
  name: pgsql-slo-alerts
    rules:
      - alert: $cityPgSQLFreshnessSLOPage
        expr: $city:job:pgsql replica replication delay is above threshold >= 1
       for: 5m
        labels:
          severity: page
         team: $city
        annotations:
          summary: "Replication Delay is High"
          description: "Replication Delay is above the threshold for 5 minutes on {{ $labels.pod }}"
      alert: $cityPgSQLFreshnessSLOWarning
```

```
expr: $city:job:pgsql_replica_replication_delay_is_above_threshold >= 1
for: lm
labels:
    severity: warning
    team: $city
annotations:
    summary: "Replication Delay is High"
    description: "Replication Delay is above the threshold for 5 minutes on {{ $labels.pod }}"
```

## Edited 1 hour ago



Write a comment or drag your files here...

Markdown is supported