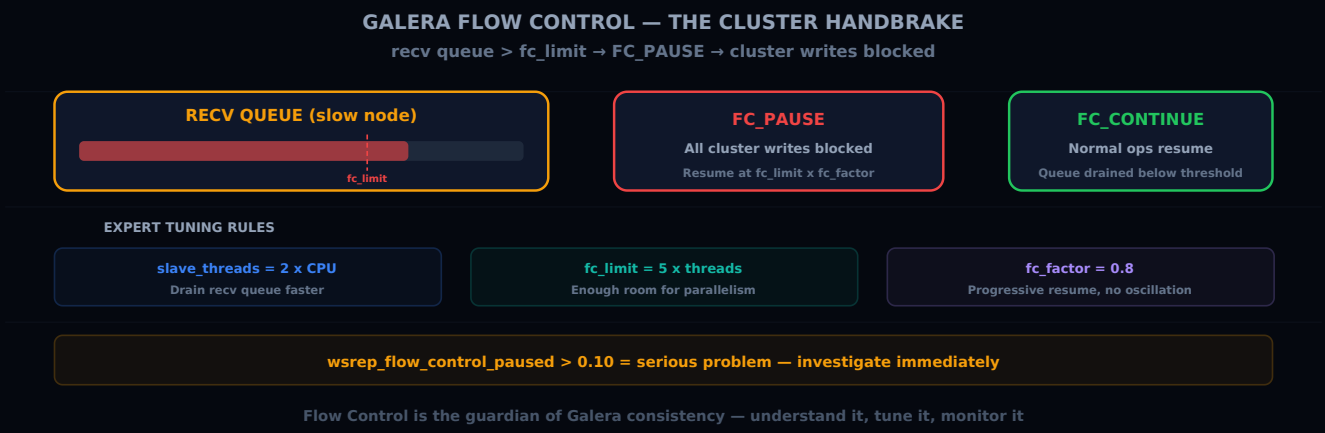
Galera: понимание Flow Control

Sylvain ARBAUDIE · October 28, 2024

GALERA MARIADB FLOW-CONTROL CLUSTERING TUNING



Фундаментальная проблема

В кластере Galera все узлы должны применять одинаковые writeset-ы в одном и том же порядке для поддержания консистентности. Но не все узлы равны: одни быстрее (новое оборудование, лёгкая нагрузка), другие медленнее (старое оборудование, тяжёлые запросы).

Что происходит, когда медленный узел не может выдержать темп записи? Writeset-ы накапливаются в его очереди приёма (recv queue). Если ничего не предпринять, эта очередь растёт неограниченно, потребляя всю память, и узел в итоге падает или отделяется от кластера.

Flow Control — это механизм, который предотвращает эту ситуацию. Это «ручной тормоз» кластера: когда узел перегружен, он просит остальных замедлиться.

Как работает Flow Control

Flow Control основан на простом пороге: **gcs.fc_limit**.

Каждый узел Galera поддерживает очередь приёма (recv queue), хранящую writeset-ы, ожидающие применения. Когда размер этой очереди превышает `gcs.fc_limit`, узел

отправляет сообщение FC_PAUSE всем остальным узлам кластера.

При получении FC_PAUSE остальные узлы прекращают отправку новых writeset-ов медленному узлу. Записи на весь кластер блокируются — это цена синхронной консистентности.

Когда recv queue медленного узла опускается ниже `gcs.fc_limit x gcs.fc_factor`, узел отправляет сообщение FC_CONTINUE и кластер возобновляет нормальную работу.

5 критических переменных wsrep

Для мониторинга Flow Control необходимы пять переменных статуса:

```
SHOW GLOBAL STATUS WHERE Variable_name IN (
    'wsrep_local_recv_queue',
    'wsrep_local_recv_queue_avg',
    'wsrep_flow_control_paused',
    'wsrep_flow_control_paused_ns',
    'wsrep_flow_control_sent'
);
```

wsrep_local_recv_queue

Текущий размер recv queue. В нормальном режиме работы это значение должно быть близко к 0. Если оно регулярно поднимается выше `gcs.fc_limit`, узел в затруднении.

wsrep_local_recv_queue_avg

Скользящее среднее recv queue. Это наиболее надёжный индикатор для выявления тенденций. Среднее выше 0,5 заслуживает расследования.

wsrep_flow_control_paused

Доля времени, проведённого в Flow Control (от 0 до 1). Если значение превышает 0,1 (10% времени), кластер имеет серьёзную проблему производительности.

wsrep_flow_control_paused_ns

Общее время в Flow Control в наносекундах. Полезно для расчёта абсолютного времени паузы за период.

wsrep_flow_control_sent

Количество отправленных сообщений FC_PAUSE этим узлом. Если один узел отправляет большинство FC_PAUSE, это узкое место, которое нужно устранить.

6 параметров тюнинга

gcs.fc_limit

Порог срабатывания Flow Control. Значение по умолчанию: 16. Увеличение этого значения допускает больший лаг перед включением тормоза, но увеличивает потребление памяти.

gcs.fc_factor

Коэффициент возобновления. Значение по умолчанию: 0,5. Когда recv queue опускается до `fc_limit x fc_factor`, Flow Control снимается. При fc_limit = 100 и fc_factor = 0,8 FC снимается при 80 writeset-ax.

wsrep_slave_threads

Количество потоков применения writeset-ов. Больше потоков = более быстрое применение полученных writeset-ов = recv queue опустошается быстрее. Рекомендация: 2 x количество ядер CPU.

wsrep_cert_deps_distance

Средняя дистанция сертификации между транзакциями. Показывает потенциальный параллелизм. Если значение высокое, увеличение `wsrep_slave_threads` даст положительный эффект.

gcs.recv_q_hard_limit

Абсолютный лимит recv queue в байтах. При превышении узел прерывается. Это последнее средство для предотвращения OOM. Рекомендация: половина RAM + swap.

gcs.max_throttle

Минимальная гарантированная пропускная способность даже в режиме Flow Control (от 0 до 1). Значение по умолчанию 0,25 означает, что даже в FC поддерживается 25%

нормальной пропускной способности. Установите в 0 для полной остановки в FC.

Экспертные рекомендации

После многих лет управления кластерами Galera в production вот консолидированные рекомендации:

Размерность slave threads

```
wsrep_slave_threads = 2 × CPU_CORES
```

Если ваш сервер имеет 8 ядер, начните с `wsrep_slave_threads = 16`. Отслеживайте `wsrep_cert_deps_distance` — если она ниже числа slave threads, уменьшите.

fc_limit в зависимости от slave threads

```
gcs.fc_limit = 5 × wsrep_slave_threads
```

При 16 slave threads `gcs.fc_limit = 80`. Это даёт потокам достаточно пространства для параллельной работы без преждевременного срабатывания FC.

fc_factor для прогрессивного возобновления

```
gcs.fc_factor = 0.8
```

fc_factor = 0,8 (вместо значения по умолчанию 0,5) обеспечивает более прогрессивное возобновление трафика, избегая осцилляций FC_PAUSE / FC_CONTINUE.

Hard limit для безопасности

```
gcs.recv_q_hard_limit = HALF_RAM_PLUS_SWAP
```

На сервере с 32 ГБ RAM и 16 ГБ swap установите `gcs.recv_q_hard_limit = 24G`. Это страховочная сетка против OOM.

Выявление проблемного узла

Когда Flow Control срабатывает часто, необходимо определить медленный узел:

```
-- На каждом узле
SELECT @@hostname,
       VARIABLE_VALUE AS fc_sent
FROM information_schema.GLOBAL_STATUS
WHERE VARIABLE_NAME = 'wsrep_flow_control_sent';
```

Узел, отправляющий наибольшее количество FC_PAUSE, является узким местом. Обычные причины:

- **Худшее оборудование**: более медленные диски, меньше RAM
- **Тяжёлые запросы**: ALTER TABLE или массивный SELECT, монополизирующий ресурсы
- **Резервное копирование**: mariabackup/xtrabackup потребляет много I/O
- **Несбалансированная нагрузка**: слишком много чтений на узле, который также должен применять writeset-ы

Заключение

Flow Control — страж консистентности в Galera. Понимание его работы и параметров необходимо для поддержания производительного кластера.

Золотые правила: `slave_threads = 2 x CPU`, `fc_limit = 5 x threads`, `fc_factor = 0.8`, hard limit = половина RAM + swap. Мониторьте `wsrep_flow_control_paused` и реагируйте, как только значение превысит 10%.

Эта статья была первоначально опубликована на Medium.