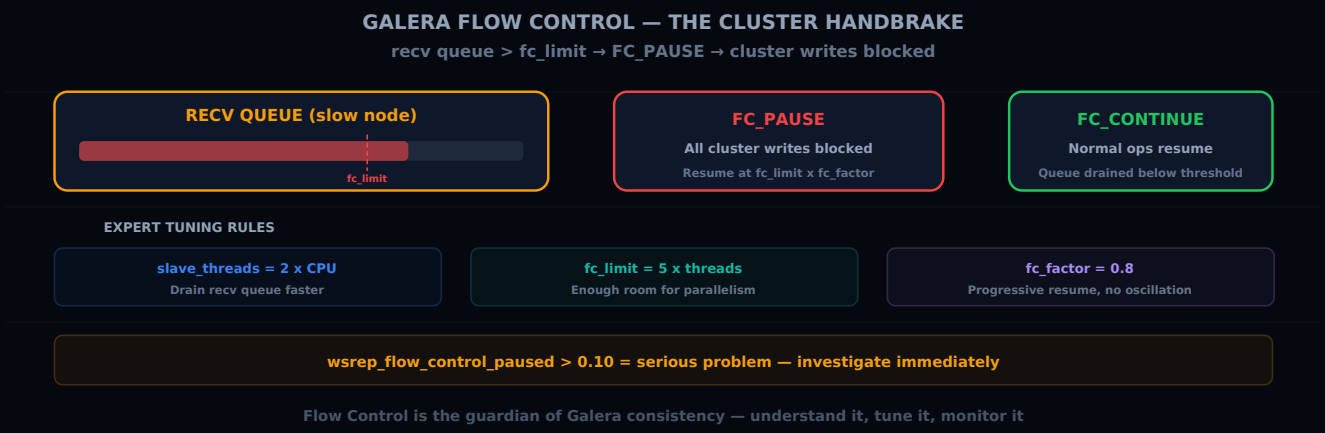
# Galera: zrozumienie Flow Control

**Sylvain ARBAUDIE** · October 28, 2024

GALERA MARIADB FLOW-CONTROL CLUSTERING TUNING




## Problem fundamentalny

W klastrze Galera wszystkie węzły muszą stosować te same writesety w tej samej kolejności, aby utrzymać spójność. Ale nie wszystkie węzły są równe: niektóre są szybsze (nowszy sprzęt, niższe obciążenie), inne wolniejsze (starszy sprzęt, ciężkie zapytania).

Co się dzieje, gdy wolny węzeł nie jest w stanie nadążyć za tempem zapisów? Writesety gromadzą się w kolejce odbiorczej (recv queue). Jeśli nic nie zostanie zrobione, kolejka rośnie nieskończenie, zużywając całą pamięć, a węzeł w końcu ulega awarii lub odchyla się od klastra.

Flow Control to mechanizm zapobiegający tej sytuacji. To hamulec ręczny klastra: gdy węzeł jest przeciążony, prosi pozostałe o zwolnienie.

## Jak działa Flow Control

Flow Control opiera się na prostym progu: **gcs.fc_limit**.

Każdy węzeł Galera utrzymuje kolejkę odbiorczą przechowującą writesety czekające na zastosowanie. Gdy rozmiar tej kolejki przekracza `gcs.fc_limit`, węzeł wysyła komunikat FC_PAUSE do wszystkich innych węzłów klastra.

Po otrzymaniu FC_PAUSE pozostałe węzły przestają wysyłać nowe writesety do wolnego węzła. Zapisy na całym klastrze są blokowane — to cena synchronicznej spójności.

Gdy recv queue wolnego węzła spadnie poniżej `gcs.fc_limit × gcs.fc_factor`, węzeł wysyła komunikat FC_CONTINUE i klaster wraca do normalnego działania.

## 5 krytycznych zmiennych wsrep

Aby monitorować Flow Control, pięć zmiennych statusowych jest niezbędnych:

```
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

Bieżący rozmiar recv queue. W normalnym działaniu ta wartość powinna być bliska 0. Jeśli regularnie przekracza `gcs.fc_limit`, węzeł ma problem.

### wsrep_local_recv_queue_avg

Średnia krocząca recv queue. Najniezawodniejszy wskaźnik do wykrywania trendów. Średnia powyżej 0,5 zasługuje na zbadanie.

### wsrep_flow_control_paused

Ułamek czasu spędzonego w Flow Control (między 0 a 1). Jeśli ta wartość przekracza 0,1 (10% czasu), klaster ma poważny problem wydajnościowy.

### wsrep_flow_control_paused_ns

Całkowity czas w Flow Control w nanosekundach. Przydatny do obliczenia absolutnego czasu pauzy w danym okresie.

### wsrep_flow_control_sent

Liczba wysłanych komunikatów FC_PAUSE przez ten węzeł. Jeśli jeden węzeł wysyła większość FC_PAUSE, to jest wąskie gardło do rozwiązania.

## 6 parametrów tuningu

### gcs.fc_limit

Próg wyzwolenia Flow Control. Wartość domyślna: 16. Zwiększenie tej wartości toleruje większe opóźnienie przed uruchomieniem hamulca, ale zwiększa zużycie pamięci.

### gcs.fc_factor

Współczynnik wznowienia. Wartość domyślna: 0,5. Gdy recv queue spadnie do `fc_limit × fc_factor`, Flow Control jest zwalniany. Przy fc_limit wynoszącym 100 i fc_factor 0,8, FC zwalnia przy 80 writesetach.

### wsrep_slave_threads

Liczba wątków stosujących writesety. Więcej wątków = szybsze stosowanie otrzymanych writesetów = recv queue opróżnia się szybciej. Rekomendacja: 2 × liczba rdzeni CPU.

### wsrep_cert_deps_distance

Średnia odległość certyfikacji między transakcjami. Wskazuje potencjalny paralelizm. Jeśli ta wartość jest wysoka, zwiększenie `wsrep_slave_threads` będzie miało pozytywny wpływ.

### gcs.recv_q_hard_limit

Absolutny limit recv queue w bajtach. Po przekroczeniu węzeł jest ubijany. To ostatnia deska ratunku przed OOM. Rekomendacja: połowa RAM + swap.

### gcs.max_throttle

Minimalny gwarantowany przepływ nawet w Flow Control (między 0 a 1). Domyślna wartość 0,25 oznacza, że nawet w FC utrzymywane jest 25% normalnego przepływu. Ustaw na 0 dla całkowitego zatrzymania w FC.

## Rekomendacje ekspertów

Po latach zarządzania klastrami Galera w produkcji, oto skonsolidowane rekomendacje:

### Wymiarowanie slave threads

```
wsrep_slave_threads = 2 × CPU_CORES
```

Jeśli Twój serwer ma 8 rdzeni, zacznij od `wsrep_slave_threads = 16`. Monitoruj `wsrep_cert_deps_distance` — jeśli jest niższa niż liczba slave threads, zmniejsz.

### fc_limit w funkcji slave threads

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

Przy 16 slave threads, `gcs.fc_limit = 80`. Daje to wystarczający margines wątkom do pracy równoległej bez zbyt wczesnego wyzwalania FC.

### fc_factor dla stopniowego wznowienia

```
gcs.fc_factor = 0.8
```

fc_factor wynoszący 0,8 (zamiast domyślnego 0,5) pozwala na bardziej stopniowe wznowienie ruchu, unikając oscylacji FC_PAUSE / FC_CONTINUE.

### Hard limit dla bezpieczeństwa

```
gcs.recv_q_hard_limit = HALF_RAM_PLUS_SWAP
```

Na serwerze z 32 GB RAM i 16 GB swap ustaw `gcs.recv_q_hard_limit = 24G`. To siatka bezpieczeństwa przed OOM.

## Identyfikacja problematycznego węzła

Gdy Flow Control wyzwala się często, trzeba zidentyfikować wolny węzeł:

```
-- Na każdym węźle
SELECT @@hostname,
       VARIABLE_VALUE AS fc_sent
FROM information_schema.GLOBAL_STATUS
```

```
WHERE VARIABLE_NAME = 'wsrep_flow_control_sent';
```

Węzeł wysyłający najwięcej FC_PAUSE to wąskie gardło. Typowe przyczyny:

- **Gorszy sprzęt**: wolniejsze dyski, mniej RAM
- **Ciężkie zapytania**: ALTER TABLE lub masowe SELECT monopolizujące zasoby
- **Trwający backup**: mariabackup/xtrabackup zużywający dużo I/O
- **Niezrównoważone obciążenie aplikacyjne**: zbyt wiele odczytów na węźle, który musi też stosować writesety

## Podsumowanie

Flow Control to strażnik spójności w Galera. Zrozumienie jego działania i parametrów jest niezbędne do utrzymania wydajnego klastra.

Złote zasady: `slave_threads = 2 × CPU`, `fc_limit = 5 × threads`, `fc_factor = 0.8`, hard limit = połowa RAM + swap. Monitoruj `wsrep_flow_control_paused` i reaguj, gdy wartość przekroczy 10%.

---

Ten artykuł został pierwotnie opublikowany na Medium.