Raport
Instytutu Automatyki i Informatyki Stosowanej
Politechniki Warszawskiej

Eksperymentalna ocena wydajności i niezawodności baz danych SQL

Ewa Niewiadomska-Szynkiewicz

E-mail: ens@ia.pw.edu.pl

Raport wykonany w ramach projektu badawczego NCN, numer 2015/17/B/ST6/01885

Warszawa, listopad 2018

SPIS TREŚCI

1. Wprowadzenie

Raport jest poświęcony zagadnieniom wydajności i niezawodności baz danych SQL. Uwaga koncentruje się na wykorzystaniu mechanizmów replikacji do zwiększenia wydajności i niezawodności baz. Prezentowane są rezultaty eksperymentów wykonanych na przykładzie baz MySQL, a konkretnie dwóch proponowanych rozwiązań: MySQL InnoDB Cluster (https://dev.mysql.com/doc/refman/8.0/en/mysql-innodb-cluster-userguide.html) oraz Percona XtraDB Cluster (https://www.percona.com/software/mysql-database/percona-xtradb-cluster). Pierwsza część raportu zawiera krótki opis systemów, sposobów ich konfiguracji oraz administracji. W drugiej części prezentowane są wyniki testów wydajnościowych i obciążeniowych oraz testów sprawdzających odporność testowanych systemów w sytuacjach wystąpienia awarii.

Wybrane technologie zostały przetestowane pod kątem odporności na awarie, wydajności oraz wpływu na działanie systemu operacyjnego. Na podstawie wyników testów wydano rekomendację dotyczącą wyboru technologii zapewniania wysokiej niezawodności baz MySQL.

2. MySQL InnoDB Cluster

MySQL InnoDB Cluster dostarcza zintegrowane natywne rozwiązanie zapewniania wysokiej niezawodności systemu bazy danych MySQL. Składa się z następujących komponentów:

- Serwery MySQL z Replikacją Grupową (ang. *Group Replication*), które zapewniają synchronizację danych we wszystkich węzłach klastra. Gwarantują wysoką odporność na awarię, udostępniają mechanizmy automatycznego przeciwdziałanie awariom oraz zwiększają elastyczność działania bazy danych.

- Router MySQL odpowiadający za równoważenie obciążenia oraz trasowanie żądań kierowanych do bazy danych przez aplikacje klienckie. W przypadku awarii jednego z serwerów żądania kierowane są do pozostałych węzłów klastra.

- Powłoka MySQL Shell, czyli narzędzie administracyjne służące do tworzenia i zarządzania klastrem InnoDB Cluster za pomocą wbudowanego interfejsu administratorskiego AdminAPI.

MySQL InnoDB to w pełni zintegrowane, autonomiczne rozwiązanie SWN dostarczane przez Oracle. Klaster działa wykorzystując sprawdzone technologie MySQL: silnik InnoDB, GTIDs (globalny identyfikator transakcji), *binary logs* (logowanie binarne), wielowątkowe węzły *slave*, replikację z wielu źródeł oraz niskopoziomowe rozwiązanie monitorowania wydajności bazy danych (Performance Schema).

3. Percona XtraDB Cluster

Percona XtraDB Cluster to skalowalne, otwarte, rozwiązanie rozszerzające standardowy serwer bazy danych MySQL w wersji 5.7 o mechanizmy zapewniania wysokiej wydajności oraz niezawodności. Jest zintegrowanym środowiskiem rozproszonym, składającym się z następujących komponentów:

- Serwer Percona MySQL, poprawiona wersją standardowego serwera MySQL w

wersji 5.7 z autorskim silnikiem bazy danych XtraDB Engine – usprawnioną wersją standardowego silnika InnoDB MySQL.

- Percona XtraBackup, nieblokujące, działające w czasie rzeczywistym narzędzie do tworzenia kopii zapasowych w systemach transakcyjnych.

- Biblioteka Codership Galera dla MySQL, dojrzałe i uznane rozwiązanie udostępniające replikację danych w bazach MySQL.

- ProxySQL, serwer pośredniczący MySQL pozwalający na rozkładanie obciążenia, trasowanie, a nawet edycję żądań kierowanych do serwera MySQL. Działa w warstwie aplikacji.

Środowisko Percona to niezależne rozwiązanie bazodanowe, będące poważną alternatywą dla natywnego oprogramowania MySQL oferowanego przez Oracle. Duża popularność Percona spowodowana była przestojem w rozwoju i niepewną przyszłością bazy danych MySQL. Obecnie, Oracle aktywnie rozwija ekosystem MySQL. Cześć usprawnień wprowadzanych przez Percona np. do silnika bazy danych wdrażana jest również w oficjalnej wersji MySQL.

4. Środowisko testowe – architektura i konfiguracja

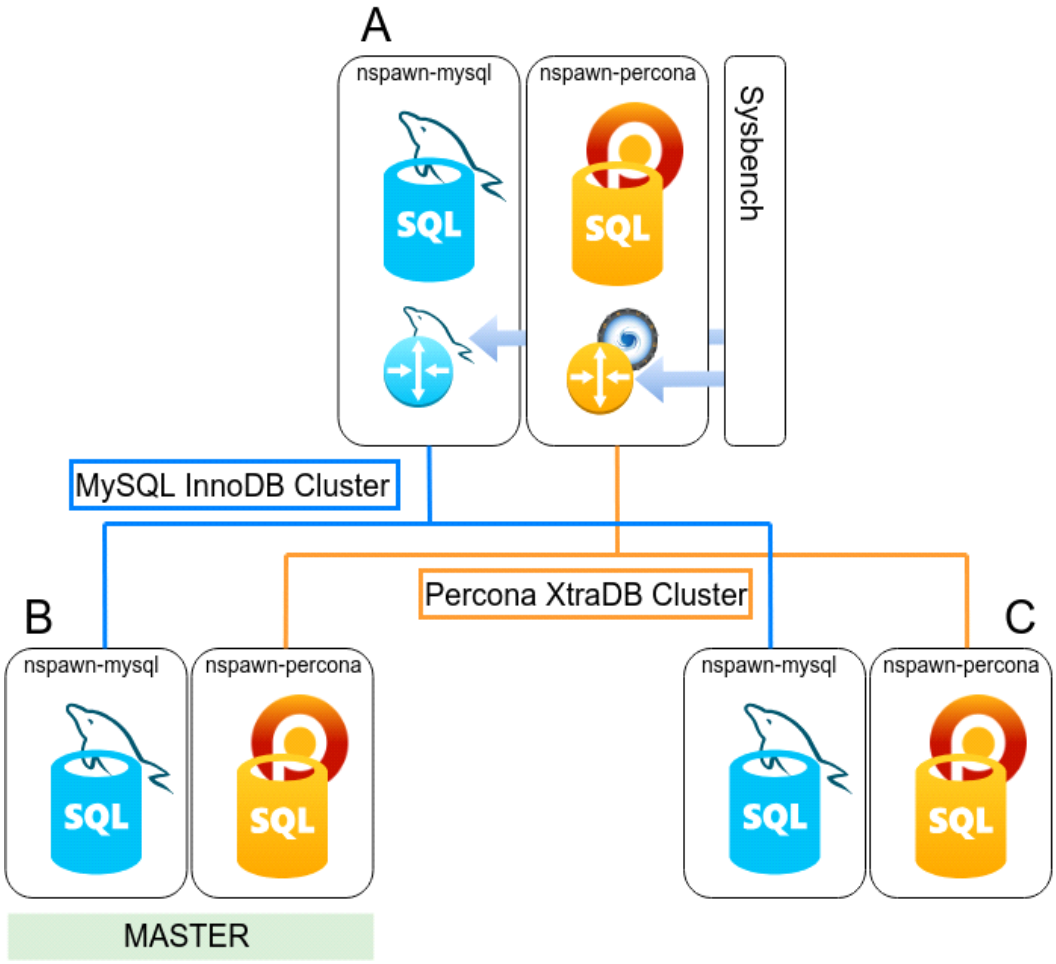
4.1 Założenia i środowisko testowe

W ramach prac badawczych przygotowano odpowiednie środowisko do prowadzenia testów. Założono, że będzie to środowisko wykorzystujące techniki wirtualizacji. Jego zadaniem miało być zasymulowanie działania rzeczywistego klastra MySQL.

Architektura i parametry środowiska testowego:

- Trzy testowe maszyny wirtualne (nazwy kodowe ***A, B, C***) połączone wydzieloną siecią prywatną.
- Parametry maszyn:
  - 8 jednostek CPU, 64 GB pamięci operacyjnej RAM, 300GB partycja na pliki bazy danych w pamięci SSD.
  - System operacyjny: Linux Debian Stretch 9.
  - przepustowość sieci VLAN klastra ok. 35 Gb/s.

Diagram prezentujący środowisko testowe z równoległą instalacją dwóch testowanych rozwiązań przedstawia rysunek 1.



Rys. 1 Środowisko testowe.

Na potrzeby testów utworzono dwa typy kontenerów Linux systemd-nspawn: mysql-nspawn i percona-nspawn z obrazami systemu Debian 9 i zainstalowanymi bazami danych, odpowiednio MySQL i Percona. Obrazy rozdystrybuowano i uruchomiono na maszynach wirtualnych ***A***, ***B*** i ***C***. Kontenery współdzielą (bez ograniczeń) zasoby oraz dziedziczą konfigurację sieciową gospodarza. Narzut związany z działaniem programu w kontenerze jest minimalny.

Serwery MySQL w kontenerach nspawn-mysql na maszynach ***A, B, C*** zostały połączone w jeden klaster MySQL InnoDB Cluster. Podobnie, serwery Percona w kontenerach nspawn-percona tworzą klaster Percona XtraDB Cluster. Oba klastry zostały skonfigurowane w trybie *single-master* – operacje zapisu danych kierowane są wyłącznie do węzła master. Węzeł główny (master) obu klastrów działa na maszynie ***B***. Ułatwia to monitorowanie stanu zasobów maszyny wirtualnej podczas testów obciążeniowych. Na węzeł główny kierowane są wszystkie zapytania utworzone przez testy. Proces testów uruchamiany jest na maszynie ***A***. Narzut związany z generowaniem zapytań testowych nie wpływa na działanie węzła głównego (***B***) baz danych. Dodatkowo na kontenerach maszyny ***A*** zainstalowano oprogramowanie do trasowania żądań do klastra bazy danych MySQL Router i ProxySQL.

4.2 Konfiguracja MySQL InnoDB Cluster

Zainstalowane oprogramowanie:

- MySQL InnoDB Cluster (v. 5.7.22)
- MySQL Router (v. 8.0.11)
- MySQL Shell (v. 8.0.11)

Paczki instalacyjne Linux Debian komponentów MySQL InnoDB Cluster są dostępne na publicznym repozytorium Debian MySQL. Konfiguracja klastra odbywa się za pomocą narzędzia MySQL Shell. Pierwszym krokiem jest przeprowadzenie lokalnej konfiguracji, każdego z węzłów klastra. Polecenie wstępnej konfiguracji wywołane jest za pomocą MySQL Shell. Zmiany zapisywane są w pliku konfiguracyjnym serwera MySQL. Dodatkowo tworzone jest konto administratora klastra bazy danych.

```
shell>      mysqlsh --uri localhost:<port>
mysqlsh[js]> dba.configureLocalInstance()
```

Na dowolnym serwerze MySQL przyszłego klastra InnoDB, należy zainicjować strukturę konfiguracyjną klastra.

```
shell>      mysqlsh --uri <admin>@<ip_1>:<port>
mysqlsh[js]> var cluster = dba.createCluster('arakis')
```

Następnie do klastra dodawane są kolejne instancje węzłów.

```
mysqlsh[js]> cluster.addInstance('<admin>@<ip_2>:<port>')
mysqlsh[js]> cluster.addInstance('<admin>@<ip_3>:<port>')
```

Jeżeli konfiguracja została przeprowadzona poprawnie oraz powiodła się synchronizacja danych między instancjami, klaster powinien zgłosić status gotowości do użycia. Kolejnym krokiem jest konfiguracja MySQL Router. W tym celu, należy uruchomić program w trybie bootstrap i podać adres dowolnego z węzłów klastra. Struktura klastra zostanie automatycznie zapisana w plikach konfiguracyjnych MySQL Router. Przy pierwszym wywołaniu zostanie utworzony użytkownik bazy danych do komunikacji router – baza danych.

```
shell> mysqlrouter --bootstrap <admin>@<ip_1>:<port> --user mysqlrouter
```

4.3 Konfiguracja Percona XtraDB Cluster

Zainstalowane oprogramowanie:

- Percona XtraDB Cluster (v. 5.7.21)
- ProxySQL (v. 1.4.8)

Paczka instalacyjna Linux Debian dla Percona XtraDB Cluster jest dostępna na publicznym repozytorium Debian Percona. Repozytorium Percona zawiera również paczkę instalacyjną ProxySQL udostępnianą na oficjalnym repozytorium ProxySQL.

Konfiguracja klastra Percona przebiega dwuetapowo. W pierwszym kroku należy dokonać

edycji plików konfiguracyjnych (np. /etc/mysql/my.cnf) serwerów MySQL, które mają należeć do klastra. Opcje związane z rozszerzeniem Galera (wsrep_*) są wstępnie wypełnione. Przykładowa konfiguracja węzła:

```
wsrep_provider          = /usr/lib/libgalera_smm.so
wsrep_cluster_name      = arakis
wsrep_cluster_address   = gcomm://192.168.70.61,192.168.70.62,192.168.70.63
wsrep_node_name         = arakis1
wsrep_node_address      = 192.168.70.61
wsrep_sst_method        = xtrabackup-v2
wsrep_sst_auth          = sstuser:passw0rd
```

Kolejnym krokiem jest uruchomienie pierwszego serwera Percona w trybie bootstrap. Pozostałe serwery, uruchamiane w trybie normalnym, powinny przyłączyć się do klastra i rozpocząć proces synchronizacji. Wszystkie dane znajdujące się na pierwszym węźle klastra zostaną automatycznie rozdystrybuowane na pozostałe węzły. W przypadku MySQL InnoDB Cluster pierwszą synchronizację trzeba wykonać ręcznie.

Podstawowa konfiguracja ProxySQL na potrzeby Percona XtraDB Cluster inicjowana jest za pomocą skryptu prosysql-admin udostępnionego przez Percone. Po w wprowadzeniu ew. zmian w pliku konfiguracyjnym proxysql-admin.conf dotyczących np. nazwy i hasła administratora klastra, wystarczy uruchomić skrypt z flagą --enable.

```
shell> proxysql-admin --config-file=/etc/proxysql-admin.cnf --enable
```

Konfiguracja ProxySQL persystowana jest na bazie danych, ewentualne poprawki można wprowadzać online, bez konieczności restartowania serwisu.

4.4 Parametry bazy danych

Serwery klastra bazy danych MySQL InnoDB oraz Percona XtraDB działają na ustawieniach domyślnych. Wyjątek stanowią parametry z poniższej tabelki. Parametry, w przypadku obu baz, zmodyfikowano na potrzeby testów.

Parametr	Wartość	Opis
max_connections	1100	Maksymalna liczba klientów łączących się jednocześnie z bazą danych
max_allowed_packet	33554432	Maksymalny dopuszczalny rozmiar pakietów jakie może wysyłać klient
max_prepared_stmt_count	10000000	Maksymalna liczba zapytań przygotowywanych jednocześnie po stronie serwera

5. Założenia do testów i scenariusze testów

5.1 Narzędzia diagnostyczne

Testy wydajnościowe, obciążeniowe oraz testy odporności na awarie zostały przeprowadzone przy użyciu narzędzia Sysbench. Aplikacja Sysbench pozwala na uruchamianie scenariuszy testowych weryfikujących działanie systemu operacyjnego, jak i baz danych, w tym MySQL. Zużycie zasobów maszyny gospodarza i stan klastra danych monitorowane było za pomocą

platformy Percona Monitoring and Management.

5.2 Testy klastrów MySQL

Klastry MySQL InnoDB Cluster i Percona XtraDB Cluster były testowane osobno. Każdy scenariusz powtarzano na obu platformach. Zestawiono i porównano wyniki odpowiednich testów. Wykonano trzy rodzaje testów:

- wydajnościowe,
- obciążeniowe,
- odporności na awarie.

6. Testy wydajnościowe

Celem testów wydajnościowych była weryfikacja efektywności działanie klastra przy różnych typach operacji bazodanowych i stopniach obciążenia. Przygotowano trzy zestawy testowe: OLTP, BULK i TPCC, każdy składający się z jednego lub więcej testów. Zestawy OLTP i BULK utworzono na podstawie istniejących scenariuszy testowych udostępnianych w ramach środowiska Sysbench. Zestaw TPCC zawiera test przygotowany przez Percone i jest dostępny w repozytorium GitHub. Test TPCC jest implementacją scenariusza TPC-C, uznanego standardu testów symulujących działanie baz typu OLTP (ang. *online transaction processing*). Wszystkie skrypty testowe zostały dostosowane do działania z klastrami baz danych i rozszerzone o dodatkowe opcje logowania wyników. Listę i krótki opis testów w zestawach testowych prezentuje Tabela 1.

Zestaw	Test	Operacje na bazie danych
OLTP	oltp_point_select	Wyszukiwanie wiersza po losowym ID
	select_random_points	Wyszukiwanie wierszy o ID z losowej tablicy
	select_random_ranges	Wyszukiwanie wierszy należących do losowych przedziałów
	oltp_update_index	Zmiana wartości komórki dla indeksowanej kolumny
	oltp_update_non_index	Zmiana wartości komórki dla nieindeksowanej kolumny
	oltp_insert	Wstawianie wiersza
	oltp_delete	Usuwanie wiersza z losowym ID
	oltp_read_only	oltp_point_select + wyszukiwanie po przedziałach i operacje na wynikach: suma, sortowanie, usuwanie powtórzeń
	oltp_write_only	oltp_update_index + oltp_update_non_index + oltp_delete
	oltp_read_write	oltp_read_only + oltp_write_only
BULK	bulk_insert	Wstawianie wielu wierszy jednocześnie
TPCC	tpcc	Operacje standardu TPC-C

Tabela 1: Testy wydajnościowe

Utworzono trzy schematy bazy danych odpowiadające zestawom testowym: OLTP, BULK oraz TPCC. Schematy OLTP i TPCC zainicjowano danymi testowymi:

- OLTP – 40 tabel po 10 000 000 wierszy (ok. 100 GB na dysku)
- TPCC – 10 zestawów danych (magazynów) w skali 100 - 90 tabel o maksymalnym rozmiarze 30 000 000 wierszy (ok 100 GB na dysku)

Podczas badań przyjęto, że węzły master klastrów działają na maszynie ***B***, testy są uruchamiane na maszynie ***A***.

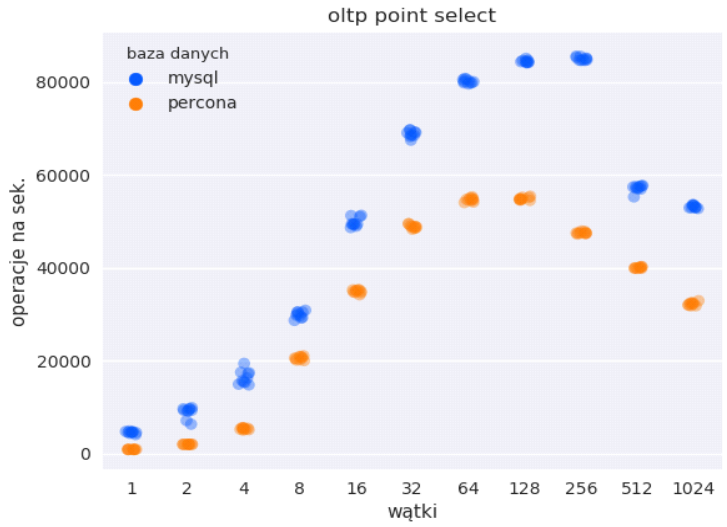
Scenariusz testów wydajnościowych

Przyjęto następujące założenia:

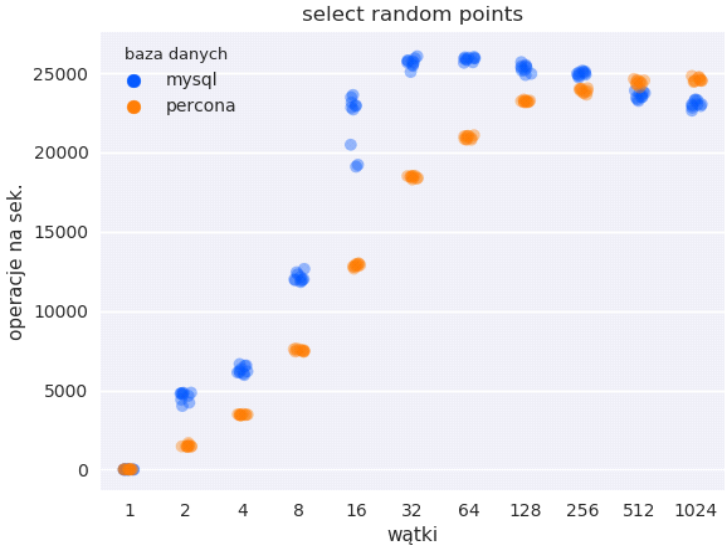
- każde zadanie testowe jest uruchamiane na kolejno 2^n wątkach $1 \leq n \leq 10$ jednocześnie,
- każdy wątek wykonuje operacje na bazie danych przez 60 sekund,
- testy są powtarzane 10 razy.
- Po przeprowadzeniu każdego testu są obliczane statystyki działania – liczba operacji bazodanowych na sekundę (transakcje, odczyty, zapisy, inne), opóźnienie operacji i liczba błędów na sekundę.

Wyniki testów

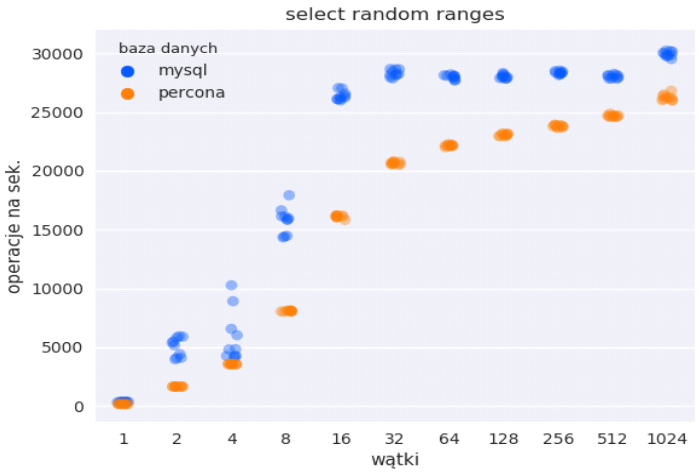
Wyniki testów zaprezentowano na rysunkach 2-11. Ich podsumowanie zawiera Tabela 2.



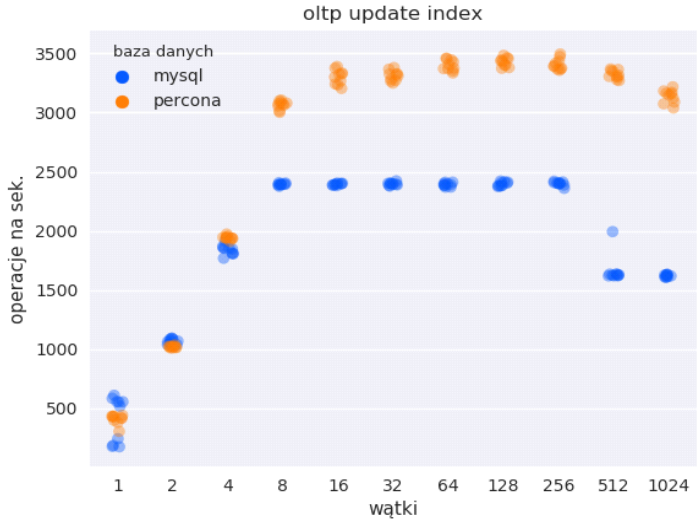
Rys. 2



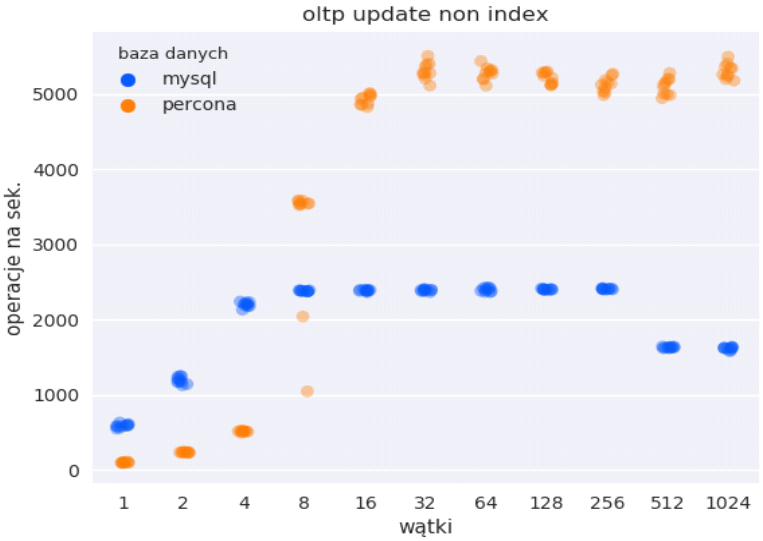
Rys. 3



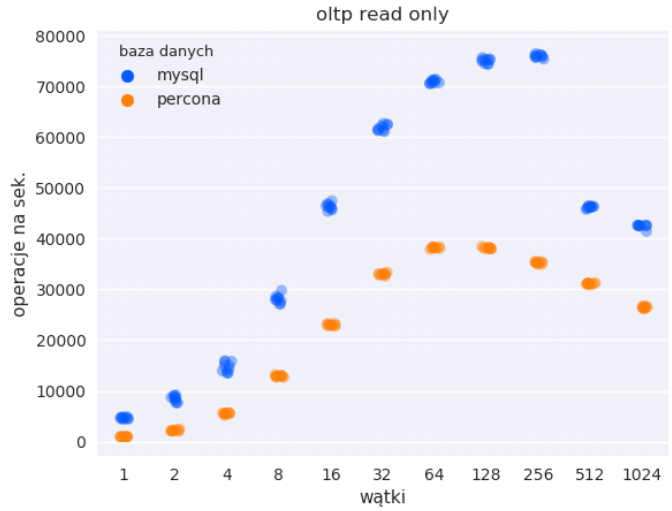
Rys 4



Rys. 5

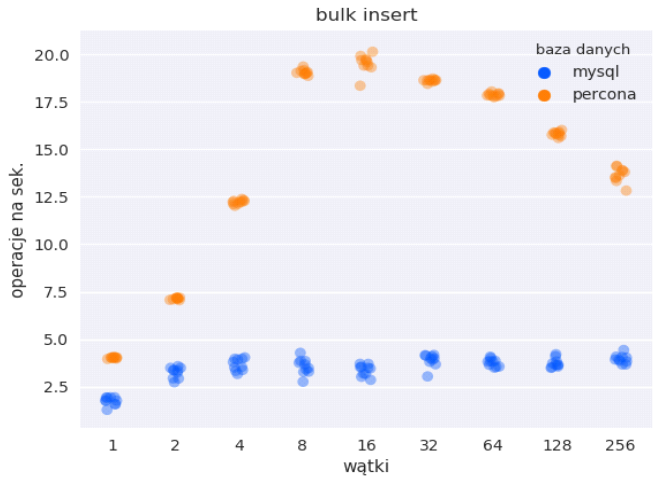


Rys. 6



Rys. 7

Rys. 8



Rys. 9

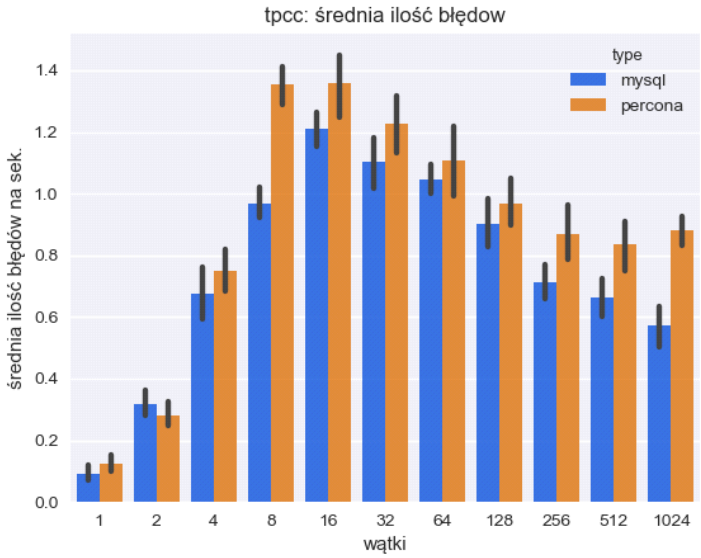
Zestaw	Test	MySQL	Percona
OLTP	*oltp_point_select*	X	
	select_random_points	X	
	select_random_ranges	X	
	oltp_update_index		X
	oltp_update_non_index		X
	oltp_insert		X
	oltp_delete	X	
	oltp_read_only	X	
	oltp_write_only	X	
	oltp_read_write	X	
BULK	*bulk_insert*		X
TPCC	*tpcc*		X
Razem:		7	5

Tabela 2: Porównanie wyników testów wydajnościowych. „X” oznacza bazę, w przypadku której operacja była wykonana szybciej.

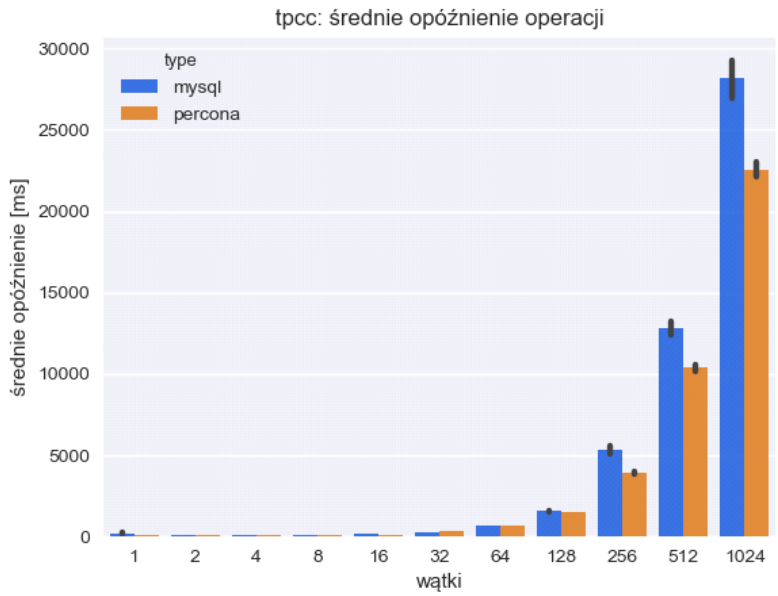
Wnioski

Wykresy oraz podsumowanie wyników w tabeli pokazują, iż MySQL InnoDB Cluster okazał się szybszy przy operacjach odczytu i usuwania wierszy z bazy danych. System Percona XtraDB Cluster osiągnął lepsze wyniki dla zapisu i edycji danych.
Największa różnica wydajności widoczna jest przy testach masowego zapisu *bulk_insert*. Na rysunku 9 widać, że MySQL InnoDB Cluster ogranicza liczbę wykonywanych operacji masowego zapisu. Szybkość zapisu nie rośnie wraz ze wzrostem liczy wątków. Zgodnie z dokumentacją, zaleca się wyłączenie procesu replikacji danych na czas operacji masowego zapisu do bazy.

W przypadku testów symulujących zróżnicowane obciążenie bazy danych - *oltp_read_write* i *tpcc*, wyniki obu rozwiązań są porównywalne. Największa rozbieżność zauważalna jest dla opóźnień operacji i liczby błędów przy testach ze znaczną liczbą wątków (256, 512, 1024). Klaster Percona charakteryzuje się mniejszym opóźnieniem operacji, ale jednocześnie większą liczbą błędów (patrz rys. 10 i 11).



Rys. 10



Rys. 11

Podsumowując, na podstawie testów wydajnościowych nie można jednoznacznie wskazać lepszego rozwiązania. Dla zrównoważonego, standardowego działania, klastry baz danych osiągają podobne wyniki. W przypadku, gdy priorytetem jest szybkość odczytu, a mniejszą wagę przykłada się do szybkości zapisu, lepszym rozwiązaniem jest MySQL InnoDB Cluster.

7. Testy obciążeniowe

Testy obciążeniowe przeprowadzono przy okazji testów wydajnościowych. Podczas wykonywania testów rejestrowano zużycie zasobów maszyny gospodarza za pomocą narzędzia Percona Management and Monitoring.

Scenariusz testów obciążeniowych

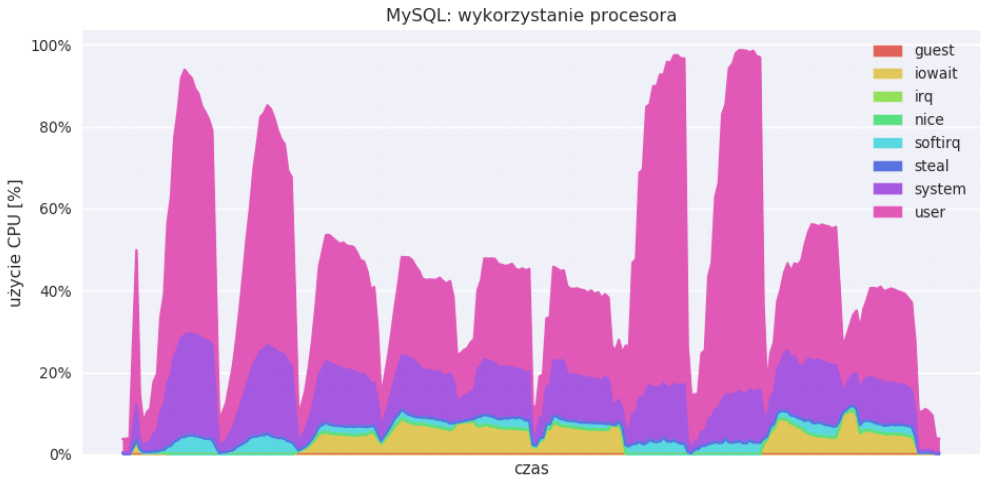
Testy wydajnościowe z zestawów OLTP wykonywano w następującej kolejności:
oltp_point_select oltp_update_index oltp_update_non_index oltp_read_only oltp_write_only oltp_read_write select_random_points select_random_ranges oltp_insert oltp_delete

Testy wykonywały się na kolejno 2^n wątkach ($1 \leq n \leq 10$) równocześnie. Obserwowano zużycie m.in. procesora, pamięci i opóźnienia operacji dyskowych. Rejestrowano ewentualne, niepożądane zachowania systemu gospodarza takie jak, wyczerpanie pamięci operacyjnej, długie opóźnienia operacji dyskowych, przedłużające się wysokie użycie czasu procesora.

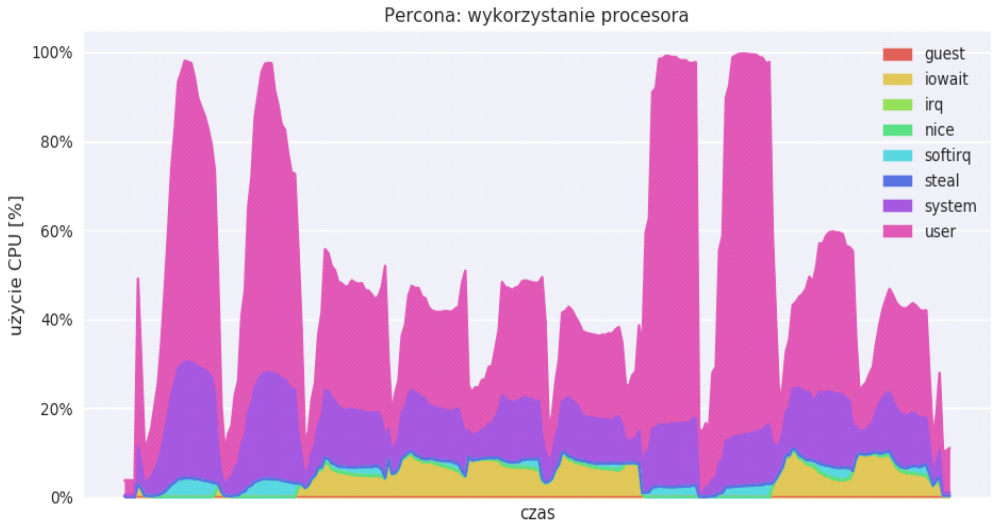
Wyniki testów obciążeniowych

Testy obciążeniowe nie wykazały żadnych niepożądanych zachowań systemu gospodarza, związanych z działaniem klastrów baz danych. Zgodnie z przewidywaniami, wykorzystanie czasu procesora znacząco rosło wraz z liczbą jednocześnie działających wątków. W

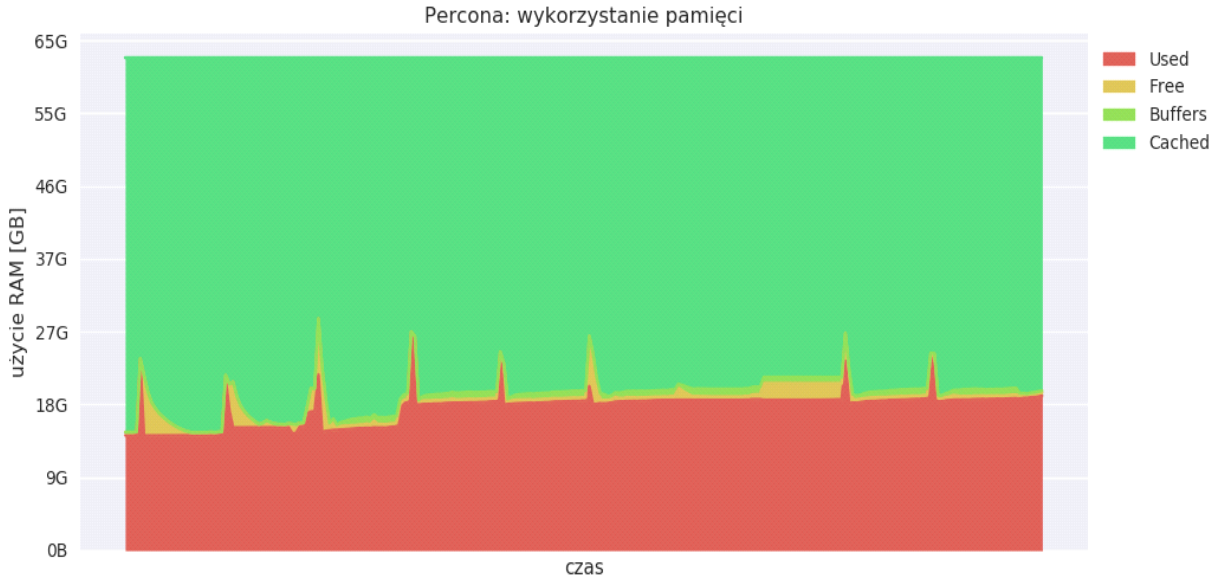
przypadku najbardziej obciążających testów (*select_random_points*, *select_random_ranges*), przy znacznej liczbie wątków, zajętość czasu procesora dochodziła nawet do 100%. Należy jednak zaznaczyć, że jest to przypadek ekstremalny. Zużycie zasobów w przypadku MySQL InnoDB Cluster i Percona XtraDB Cluster jest porównywalne. Z rysunków 12-14 wynika, że węzły klastra MySQL w mniejszym stopniu zużywają czas procesora. Jednocześnie, węzły klastera Percona alokują średnio mniej pamięci operacyjnej. Różnice są jednak nieznaczne.



Rys. 12



Rys. 13



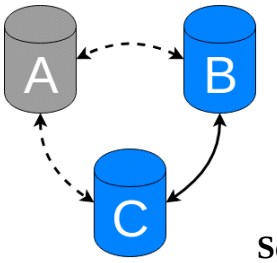
Rys. 14

Podsumowanie
Na podstawie przeprowadzonych testów widać, że obie rozważane technologie w równym stopniu obciążają system operacyjny.

8. Testy odporności na awarie

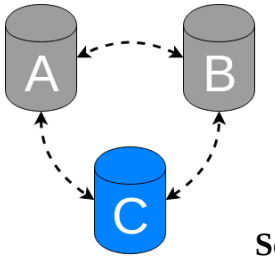
Scenariusze testów odpornościowych

Testy odporności na awarię przeprowadzono symulując możliwe scenariusze awarii węzłów klastra bazy danych. W każdym przypadku uruchamiano test *oltp_read_write*, symulujący normalne działania prowadzane na bazie danych. Badano odporność obu technologii: MySQL InnoDB Cluster i Percona XtraDB Cluster. Po każdym teście realizowano odpowiednią dla danej technologii, procedurę naprawczą. Po zakończeniu procedury weryfikowano poprawne działanie klastra i spójność danych. Rozpatrywano siedem scenariuszy awarii:



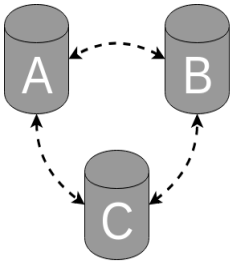
Scenariusz 1.

Węzeł ***A*** został poprawnie zatrzymany. Klaster kontynuuje swoje działanie, ale przestaje być odporny na awarię jednego z węzłów. Po ponownym włączeniu, węzeł ***A*** powinien ponownie dołączyć się do klastra i zsynchronizować dane.



Scenariusz 2.

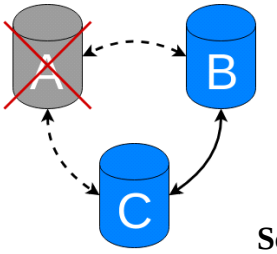
Węzły ***A*** i ***B*** zostały poprawnie zatrzymane. Podobnie jak w scenariuszu 1, ale dla dwóch węzłów. Klaster jest nieodporny na awarie, jeden pozostały węzeł powinien odpowiadać na żądania klientów. Po ponownym uruchomieniu węzłów ***A*** i ***B***, powinna istnieć możliwość odbudowania klastra.



Text

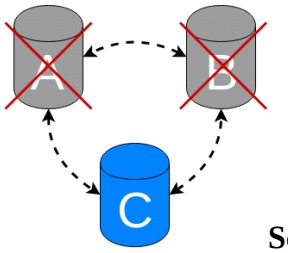
Scenariusz 3.

Wszystkie węzły zostały poprawnie zatrzymane. Klaster został wyłączony. Po ponownym uruchomieniu maszyn, klaster można odtworzyć, bez utraty spójności danych.



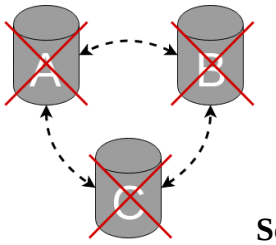
Scenariusz 4.

Węzeł ***A*** nie odpowiada. Węzeł z niewyjaśnionych przyczyn utracił połączenie z klastrem, np. z powodu braku zasilania, problemów sprzętowych, błędów jądra, błędów procesu serwera bazy danych. Dwa pozostałe węzły próbują nawiązać połączenie z węzłem ***A***. W przypadku niepowodzenia węzeł zostanie usunięty z klastra. Kworum zostaje zachowane (2 z 3), klaster nieprzerwanie pozwala na dostęp i edycję danych. Po uruchomieniu węzła A, powinna istnieć możliwość ponownego włączenia go do klastra jak w scenariuszu 1.



Scenariusz 5.

Węzły ***A*** i ***B*** nie odpowiadają. Utracono kworum (1 z 3), węzeł ***C*** nie powinien zezwolić na zapis danych do bazy ze względu na możliwość wystąpienia konfliktów danych. Odbudowanie klastra jest możliwe po uruchomieniu węzłów ***A*** i ***B*** oraz ich poprawnej synchronizacji.



Scenariusz 6.

Nie odpowiada żaden węzeł klastra. Wszystkie węzły opuściły klaster bez przeprowadzenia poprawnie zakończonej procedury wyjścia. Scenariusz może mieć miejsce np. w przypadku globalnej awarii zasilania. Procedura naprawy klastra wymaga identyfikacji najstarszego węzła - zawierającego aktualne dane, następnie synchronizacji pozostałych węzłów względem najstarszego węzła i ponownego połączenia ich w klaster bazy danych.

Wyniki testów

Przeprowadzone badania wykazały, że oba testowane rozwiązania zapewniają wymagany poziom odporności na awarie. Procedury przywracania stanu systemu różnią się w zależności od technologii. Tabela 3 prezentuje możliwości automatycznej synchronizacji klastra dla każdego ze scenariuszy.

	Automatyczna synchronizacja	
Scenariusz	MySQL	Percona
1	✓/✗	✓
2	✓/✗	✓
3	✗	✓
4	✓/✗	✓/✗
5	✗	✓/✗
6	✗	✗

Tabela 3: Możliwość automatycznej synchronizacji dla scenariusza awarii.

Podsumowanie

Percona XtraDB Cluster oferuje lepszy system synchronizacji danych, wspiera dwie metody replikacji:

- IST – (ang. *incremental state transfer*) Przyrostowy transfer stanu, lekka synchronizacja różnicowa – wysyłane są wyłącznie dane różniące się na obu instancjach.
- full SST – (ang. *state snapshot transfer*) Pełny transfer stanu – na węźle docelowym replikowany jest cały obraz węzła wzorcowego.

Synchronizacja zwykle przebiega w sposób automatyczny. Zarządzanie klastrem odbywa się za pomocą plików konfiguracyjnych i parametrów startowych instancji klastra. Stan klastra dostępny jest przez interfejs bazodanowy. Status replikacji zapisywany jest w logach.

MySQL InnoDB Cluster kładzie większy nacisk na kontrolę i rozproszenie konfiguracji klastra. Ustawienia początkowe inicjowane są na podstawie plików konfiguracyjnych. Zarządzanie klastrem odbywa się za pomocą narzędzia MySQL Shell, które pozwala na

sprawdzanie statusu klastra oraz zmiany jego topologii i konfiguracji węzłów. W przypadku awarii takiej jak np. dłuższa nieobecność węzła w klastrze, wymagane jest wywołanie odpowiednich procedur za pomocą MySQL Shell.

9. Podsumowanie i wnioski

Wykonano badania porównawcze dwóch konkurencyjnych systemów wysokiej niezawodności baz danych MySQL: MySQL InnoDB Server i Percona XtraDB Cluster. Zweryfikowano ich wydajność, poziom zużywanych zasobów obliczeniowych oraz odporność na awarie. Wyniki testów działania klastrów są porównywalne. Główna różnica między rozwiązaniem MySQL i Percona to sposób realizacji operacji replikacji. W przypadku Percony mechanizm jest bardziej rozbudowany i kładzie większy nacisk na automatyzację rozwiązywania konfliktów synchronizacji. Z drugiej strony system MySQL wymaga większego udziału administratora w zarządzaniu klastrem, ale udostępnia wygodne narzędzie administracji jakim jest MySQL Shell.

Zalety Percona XtraDB Cluster:

- Dojrzalsze, lepiej przetestowane rozwiązanie.
- Szybszy zapis danych do bazy.
- Większa automatyzacja przy naprawie awarii.
- Wysokiej jakości dokumentacja.
- Wygodna integracja z narzędziem do monitorowania baz danych Percona Management and Monitoring.

Zalety MySQL InnoDB Cluster:

- Oficjalny produkt dostawcy oprogramowania Oracle.
- Szybszy odczyt z bazy danych.
- Wygodne narzędzie do administracji klastrem baz danych MySQL Shell.