O'REILLY®

Wydanie IV

Wysoko wydajne MySQL

Sprawdzone strategie działania na dużą skalę

Silvia Botros
Jeremy Tinley

Helion

Tytuł oryginału: High Performance MySQL: Proven Strategies for Operating at Scale, 4th Edition

Tłumaczenie: Robert Górczyński

ISBN: 978-83-283-9294-6

© 2022 Helion S.A.

Authorized Polish translation of the English *High Performance MySQL 4E*
ISBN 9781492080510 © 2022 Silvia Botros and Jeremy Tinley.

This translation is published and sold by permission of O'Reilly Media, Inc.,
which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/wywym4
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Kup książkę Poleć książkę

ROZDZIAŁ 2.

Monitorowanie w świecie inżynierii niezawodności

Systemy monitorowania to rozległy temat, który w ciągu kilku minionych lat został mocno ukształtowany przez pozycje takie jak *Site Reliability Engineering. Jak Google zarządza systemami produkcyjnymi* (Helion) i *The Site Reliability Workbook: Practical Ways to Implement SRE* (O'Reilly). Od chwili wydania tych dwóch książek inżynieria niezawodności (ang. *site reliability engineering*, SRE) stała się popularnym trendem w ofertach pracy. Niektóre firmy posunęły się nawet do zmiany nazw pewnych stanowisk na różne wariacje „inżynierii niezawodności".

Inżynieria niezawodności doprowadziła do zmiany sposobu, w jaki w zespołach myśli się o pracy operacyjnej. Wynika to stąd, że na inżynierię niezawodności składa się wiele zbiorów reguł, pozwalających znacznie łatwiej udzielić odpowiedzi na następujące pytania:

- Czy użytkownikowi jest zapewniany odpowiedni poziom wrażeń?
- Czy należy skoncentrować się na niezawodności i odporności na awarie?
- Jak zrównoważyć nowe funkcje z wysiłkiem potrzebnym do ich opracowania?

W tym rozdziale przyjęliśmy założenie, że czytelnik w pełni rozumie te reguły. Jeżeli czytelnik nie przeczytał żadnej z wymienionych książek, jako krótkie wprowadzenie do tematu inżynierii niezawodności zalecamy lekturę następujących rozdziałów książki *The Site Reliability Workbook*:

- rozdział 1., w którym dokładnie omówiono filozofię kryjącą się za ruchem w kierunku zarządzania wydajnością działania na poziomie usługi w środowisku produkcyjnym;
- rozdział 2., w którym wyjaśniono, jak implementować cele na poziomie usługi (ang. *service level objective*, SLO);
- rozdział 5., w którym pokazano, jak modyfikować cele SLO.

Niektórzy mogą twierdzić, że implementacja SRE nie jest ściśle powiązana z wysoką wydajnością MySQL, ale my się z tym nie zgadzamy. W swojej książce *Przyspieszenie. Lean i DevOps w rozwoju firm technologicznych*[1] dr Nicole Forsgren twierdzi, że należy skoncentrować się na wynikach, a nie na konkretnych wartościach. Kluczowym aspektem efektywnego zarządzania serwerem MySQL

[1] Nicole Forsgren, *Przyspieszenie. Lean i DevOps w rozwoju firm technologicznych*, Helion (*https://helion.pl/ksiazki/przyspieszenie-lean-i-devops-w-rozwoju-firm-technologicznych-nicole-forsgren-phd-jez-humble-gene-kim,przyld.htm*).

Kup książkę Poleć książkę

jest dobre monitorowanie kondycji baz danych. Tradycyjne monitorowanie można porównać do dobrze znanej drogi. Z kolei inżynieria niezawodności to nowa droga, mniej znana i rozumiana pod tym względem, jak należy implementować reguły SRE w bazie danych MySQL. Wraz z coraz szerszym poziomem akceptacji reguł inżynierii niezawodności będzie ewoluowała tradycyjna rola administratora baz danych — administratorzy będą musieli uwzględniać także sposoby monitorowania systemów, którymi zarządzają.

Wpływ inżynierii niezawodności na zespoły administratorów baz danych

Przez wiele lat monitorowanie wydajności działania bazy danych koncentrowało się na zagłębieniu w wydajność działania pojedynczego serwera. Wprawdzie nadal ma to dużą wartość, ale dotyczy pomiarów reaktywnych, np. podczas profilowania serwera charakteryzującego się kiepską wydajnością działania. To była standardowa procedura operacyjna w czasie, gdy tylko zespoły administratorów baz danych odpowiadały za sposób działania bazy danych i nikt inny nie miał dostępu do tych informacji.

Witaj w świecie opracowanej przez Google filozofii inżynierii niezawodności. Rola administratora baz danych stała się bardziej złożona i podążyła bardziej w stronę inżynierii niezawodności witryny internetowej (SRE) lub inżynierii niezawodności bazy danych (DBRE). Zespoły musiały zoptymalizować sposoby wykorzystania swojego czasu. Poziomy usługi umożliwiają zdefiniowanie, kiedy klienci są niezadowoleni, co pozwala na lepszy podział czasu między zajmowanie się kwestiami dotyczącymi wydajności działania i wyzwaniami związanymi ze skalowalnością a pracę nad wewnętrznymi narzędziami. Omówimy różne sposoby monitorowania serwera MySQL, aby zapewnić jak najlepsze wrażenia jego użytkownikom.

Definiowanie celów poziomu usługi

Zanim zagłębimy się w pomiar zadowolenia klientów związany z wydajnością działania klastrów bazy danych, najpierw trzeba ustalić cele i zdefiniować je zwykłym językiem. Oto kilka pytań, które w organizacji można potraktować jako punkt wyjścia do zdefiniowania tych celów:

- Czy istnieją wskaźniki odpowiednie do pomiaru sukcesu?
- Jakie wartości tych wskaźników są akceptowane dla klientów i potrzeb biznesowych?
- W jakich okolicznościach mamy do czynienia ze stanem zdegradowanym?
- Kiedy znajdujemy się w stanie awarii wymagającym jak najszybszego działania?

Istnieją scenariusze z oczywistymi odpowiedziami na te pytania (np. baza danych źródła nie działa, nie można akceptować operacji zapisu, a tym samym cele biznesowe nie są realizowane). Inne scenariusze są mniej oczywiste, np. okresowo wykonywane zadanie czasami zabiera bazie danych wszystkie dyskowe operacje wejścia-wyjścia i nagle wszystko działa bardzo wolno. Zrozumienie w organizacji tego, co jest mierzone i dlaczego, może pomóc w nadawaniu odpowiednich poziomów ważności zadaniom. Osiągnięcie tego stanu przez nieustanny dialog w organizacji pomaga

Kup książkę Poleć książkę

ustalić, czy wysiłek inżynierów powinien być skierowany na nową funkcjonalność, czy raczej potrzebna jest większa inwestycja w poprawienie wydajności działania lub stabilności.

W filozofii SRE takie dyskusje dotyczące zadowolenia klienta naprowadzą zespół na to, co jest dobre dla organizacji w kategoriach wskaźników poziomu usługi (ang. *service level indicator*, SLI), celów na poziomie usługi (ang. *service-level objective*, SLO) i umów na poziomie usługi (ang. *service level agreement*, SLA).

Wskaźnik poziomu usługi (SLI)

Ujmując rzecz najprościej: SLI pozwala odpowiedzieć na pytanie „jak można sprawdzić, czy klienci są zadowoleni?". Odpowiedź przedstawia „zdrowy" system z perspektywy użytkownika. SLI może być wskaźnikiem biznesowym typu „czas udzielenia odpowiedzi dla API używanego przez klienta" lub bardziej ogólnym „czas działania usługi". Być może wystąpi konieczność użycia odmiennych wskaźników w zależności od kontekstu danych i jego związku z produktem.

Cel na poziomie usługi (SLO)

SLO pozwala odpowiedzieć na pytanie „jaka jest minimalna postać SLI, aby klienci byli zadowoleni?". Wartość tego wskaźnika można uznać za obiektywny zakres SLI, w którym należy się znaleźć, aby usługa mogła być uznana za zdrową. Jeżeli za wartość SLI przyjmiemy czas nieprzerwanej pracy, to liczba dziewiątek określających ten czas będzie wartością SLO. Wskaźnik SLO można zdefiniować jako wartość *na przestrzeni danego czasu*, aby zagwarantować, że każdy członek zespołu będzie rozumieć znaczenie SLO. Dzięki połączeniu SLO i SLI otrzymujemy podstawowe równanie pomocne w ustaleniu, czy klienci są zadowoleni.

Umowa na poziomie usługi (SLA)

SLA pozwala udzielić odpowiedzi na pytanie „na jaką wartość SLO jestem w stanie się zgodzić, gdy wiążą się z tym pewne konsekwencje?". SLA można uznać za wskaźnik SLO ujęty w umowie z co najmniej jednym klientem biznesowym (czyli z klientem wnoszącym opłaty, a nie z udziałowcem), zgodnie z którą niedotrzymanie warunków SLA oznacza kary finansowe. Trzeba w tym miejscu dodać, że wskaźnik SLA to kwestia opcjonalna.

W tym rozdziale nie będziemy się zbytnio koncentrowali na SLA, ponieważ związane z nim zagadnienia są bardziej dyskusją biznesową niż techniczną. Ten rodzaj decyzji zależy głównie od oczekiwanej sprzedaży, gdy w kontrakcie będzie zawarta wartość wskaźnika SLA, oraz od tego, czy jest to warte ryzyka utraty zysku w przypadku niedotrzymania warunków SLA. Należy mieć nadzieję, że taka decyzja jest podejmowana świadomie, na podstawie zamieszczonych tutaj informacji dotyczących wskaźników SLI i SLO.

Zdefiniowanie wartości SLI, SLO i SLA ma wpływ nie tylko na kwestie biznesowe przedsięwzięcia, ale również na planowanie w zespole inżynierów. Jeżeli zespołowi nie uda się osiągnąć ustalonej wartości wskaźnika SLO, nie powinien rozpoczynać pracy nad nową funkcjonalnością. To samo dotyczy zespołów odpowiedzialnych za obsługę bazy danych. Jeżeli jedna z potencjalnych wartości SLO omawianych w tym rozdziale nie została spełniona, powinno to być bodźcem do dyskusji o powodach takiego stanu rzeczy. Jeżeli uzbroisz się w dane pomagające wyjaśnić, dlaczego wrażenia użytkownika nie są optymalne, to masz szanse na znacznie bardziej owocną komunikację z zespołem na temat priorytetów.

Kup książkę Poleć książkę

Co spowoduje zadowolenie klientów?

Po wyborze zestawu wskaźników dla SLI kuszące może być zdefiniowanie celów jako osiągnięcia wartości 100%. Trzeba będzie oprzeć się tej pokusie. Należy pamiętać, że wybór wskaźników i celów ma być oceniony w każdej chwili, z użyciem obiektywnych wskaźników, niezależnie od tego, czy zespół może zająć się opracowywaniem nowych funkcji, czy też musi poświęcić więcej uwagi i zasobów na dopracowanie istniejącej funkcjonalności ze względu na ryzyko spadku jakości poniżej poziomu akceptowanego przez klientów. Celem jest zdefiniowanie *absolutnego minimum*, jakie trzeba osiągnąć, aby klienci byli zadowoleni. Jeżeli klient będzie zadowolony, gdy na stronę docelową trafi w ciągu 2 sekund, nie ma potrzeby definiowania celu polegającego na wczytaniu strony w 750 milisekund, gdyż może to spowodować nieuzasadnione obciążenie w zespole inżynierów.

Wykorzystując przykład czasu nieustannego działania jako wskaźnika i obiektywnych wartości dla niego, można zadeklarować „brak jakiegokolwiek czasu przestoju". Jakie to ma znaczenie podczas implementowania i śledzenia, czy cele zostały spełnione? Osiągnięcie poziomu dostępności wyrażonego trzema dziewiątkami jest nie lada wyczynem. Te trzy dziewiątki w trakcie całego roku oznaczają nieco ponad 8 godzin niedostępności usługi, czyli mniej więcej 10 minut tygodniowo. Im więcej dziewiątek będzie obiecanych, tym trudniej będzie spełnić obietnicę i większy będzie koszt z tym związany. W tabeli 2.1 zamieściliśmy pochodzącą z usługi Amazon Web Services pomocną tabelę, która pokazuje te wyzwania wyrażone liczbami.

Tabela 2.1. Dostępność podawana w dziewiątkach

Dostępność	Roczny czas przestoju	Miesięczny czas przestoju	Tygodniowy czas przestoju	Dzienny czas przestoju
99,999%	5 minut, 15,36 sekundy	26,28 sekundy	6,06 sekundy	0,14 sekundy
99,995%	26 minut, 16,8 sekundy	2 minuty, 11,4 sekundy	30,3 sekundy	4,32 sekundy
99,990%	52 minuty, 33,6 sekundy	4 minuty, 22,8 sekundy	1 minuta, 0,66 sekundy	8,64 sekundy
99,950%	4 godziny, 22 minuty, 48 sekund	31 minut, 54 sekundy	5 minut, 3 sekundy	43 sekundy
99,900%	8 godzin, 45 minut, 36 sekund	43 minuty, 53 sekundy	10 minut, 6 sekund	1 minuta, 26 sekund
99,500%	43 godziny, 48 minut, 36 sekund	3 godziny, 39 minut	50 godzin, 32 minuty, 17 sekund	7 minut, 12 sekund
99,250%	65 godzin, 42 minuty	5 godzin, 34 minuty, 30 sekund	1 godzina, 15 minut, 48 sekund	10 minut, 48 sekund
99,000%	3 dni, 15 godzin, 54 minuty	7 godzin, 18 minut	1 godzina, 41 minut, 5 sekund	14 minut, 24 sekundy

Ponieważ czas inżynierów to ograniczony zasób, trzeba pamiętać, aby nie dążyć do perfekcji podczas wyboru SLO. Nie każda funkcjonalność produktu będzie wymagała wszystkich dziewiątek w celu zadowolenia klienta. Dlatego też wraz ze wzrostem zbioru funkcjonalności produktu okazuje się, że wskaźniki SLI i SLO są zróżnicowane, w zależności od ich wpływu na określoną funkcjonalność bądź na osiągany z niej zysk. To jest oczekiwane i świadczy o przemyślanym procesie. W tym miejscu trzeba wykonać zadanie o znaczeniu krytycznym: ustalić, kiedy zbiór danych

Kup książkę Poleć książkę

staje się zagrażającym wydajności działania wąskim gardłem dla różnych profili zapytań wykonywanych przez różnych użytkowników. To oznacza również znalezienie sposobu na rozdzielenie potrzeb różnych użytkowników, aby można było zapewnić im rozsądne wartości SLI i SLO.

Te wskaźniki i cele to także efektywny sposób na przygotowanie jednolitego języka stosowanego między produktem i inżynierami, pomagającego w podejmowaniu decyzji typu „praca nad nową funkcjonalnością" kontra „poświęcenie czasu na usunięcie problemów i zapewnienie większej niezawodności". To również sposób na określenie, które z elementów listy mają zostać zaimplementowane i które są najważniejsze, na podstawie wrażeń klientów. Wartości SLI i SLO można wykorzystać do ustalania hierarchii ważności podczas komunikacji w zespole, która w innych przypadkach byłaby trudna do uzgodnienia.

Co należy mierzyć?

Wyobraźmy sobie firmę, której produktem jest sklep internetowy. W firmie zauważono większy ruch sieciowy ze względu na większą liczbę klientów odwiedzających sklep w internecie. Pojawiło się zapotrzebowanie na grupę infrastruktury w celu zagwarantowania, że warstwa bazy danych będzie w stanie obsłużyć zwiększone zainteresowanie firmą. W tym podrozdziale wyjaśnimy, co powinno być mierzone przez zespół zajmujący się infrastrukturą.

Definiowanie SLI i SLO

Zdefiniowanie dobrej wartości SLI i dopasowanej SLO koncentruje się na zwięzłym wyjaśnieniu, jak zapewnić wspaniałe wrażenia klientom. Nie będziemy w tym miejscu poświęcać więcej czasu na omówienie sposobów tworzenia rozsądnych SLI i SLO[2]. W kontekście MySQL to musi być reprezentacja definiująca trzy główne aspekty: dostępność, opóźnienie i brak błędów o znaczeniu krytycznym.

W przypadku wspomnianego wcześniej sklepu internetowego oznacza to, że strony są wczytywane szybko, w czasie krótszym niż kilkaset milisekund w co najmniej 99,5% przypadków mierzonych na przestrzeni czasu. To oznacza również niezawodny proces finalizacji zamówienia, w którym sporadyczne awarie są dozwolone jedynie przez 1% czasu w danym miesiącu kalendarzowym. Nie definiujemy celu w postaci wartości 100%, ponieważ funkcjonujemy w świecie, w którym awarie są nieuchronne. Korzystamy z przedziału czasu, aby zespół mógł właściwie równoważyć swoją pracę między opracowywaniem nowych funkcjonalności i zwiększaniem odporności produktu na awarie.

„Oczekuję, że 99,5% zapytań do bazy danych będzie bez błędów spełnione w czasie krótszym niż 2 milisekundy" to wystarczająca definicja SLI i przejrzysta definicja SLO, choć jednocześnie niełatwa do spełnienia. Nie można tego wszystkiego potwierdzić za pomocą pojedynczego wskaźnika. To zdanie wyraża oczekiwany sposób działania warstwy bazy danych w celu dostarczenia akceptowanego poziomu wrażeń użytkownika.

[2] Gorąco polecamy pozycję *Implementing Service Level Objectives* Alexa Hidalgo (O'Reilly).

Kup książkę Poleć książkę

Jaki jest więc we wspomnianym wcześniej sklepie internetowym dobry przykład wskaźników, na podstawie których można tworzyć obraz wrażeń użytkownika? Należy rozpocząć od syntetycznych testów, takich jak wczytywanie strony w środowisku produkcyjnym, z przykładową szybkością wczytywania. Będzie to użytecznym sygnałem wskazującym, że „wszystko jest w porządku". Jednak to dopiero jest początek. Zapoznaj się z innymi aspektami sygnałów, które należy śledzić, aby otrzymać pełny obraz sytuacji. W miarę poruszania się po tych przykładach powiążemy je ze wspomnianym sklepem internetowym, aby w ten sposób pomóc zwizualizować, jak poszczególne wskaźniki pozwalają utworzyć obraz dobrych wrażeń klienta. Przede wszystkim trzeba zająć się śledzeniem czasu udzielenia odpowiedzi.

Rozwiązania w zakresie monitorowania

Analiza zapytania i monitorowanie opóźnienia zapytania w kontekście SLI i SLO muszą koncentrować się na wrażeniach klienta. To oznacza bazowanie na narzędziach, które mogą jak najwcześniej alarmować, gdy czas udzielenia odpowiedzi jest dłuższy niż ustalona wartość progowa. Warto przeanalizować kilka ścieżek, dzięki którym można osiągnąć ten poziom monitorowania.

Rozwiązania komercyjne

To jest jeden z przykładów, gdy zakup produktu, którego przewaga nad konkurencją wiąże się z konkretnym aspektem profilowania wydajności działania MySQL, może okazać się korzystny dla organizacji. Narzędzia takie jak SolarWinds Database Performance Management (*https://www.solarwinds.com/solutions/database-solutions*) mogą zapewnić automatyzację profilowania wydajności działania zapytania, a także udostępnić tę funkcjonalność większej liczbie inżynierów w organizacji.

Rozwiązania open source

Doskonale znanym rozwiązaniem open source jest Percona Monitoring and Management (*https://docs.percona.com/percona-monitoring-and-management/index.html*), czyli PMM. Oprogramowanie to działa jako para klient-serwer. Klient jest instalowany w egzemplarzach baz danych, które pobierają wskaźniki i przekazują je do serwera. Z kolei serwer zawiera zbiór paneli razem z wykresami pokazującymi w sposób graficzny wydajność działania. Jedną z największych zalet PMM jest ułożenie paneli, wybrane na podstawie dużego doświadczenia w społeczności Percony związanej z monitorowaniem wydajności działania MySQL. W ten sposób powstało doskonałe źródło, które początkującym inżynierom MySQL pomaga w monitorowaniu wydajności działania MySQL.

Inne rozwiązanie polega na przekazaniu dziennika wolno wykonywanych zapytań i danych wyjściowych MySQL Performance Schema do scentralizowanego położenia, w którym doskonale znane narzędzia, takie jak będące częścią pakietu Percona Toolkit *pt-query-digest*, zostaną użyte do analizowania dzienników zdarzeń i zebrania dokładniejszych informacji o działaniach wykonywanych przez egzemplarze baz danych. Wprawdzie ten proces jest efektywny, ale jednocześnie może być wolny, a jego niepoprawne zastosowanie będzie miało wpływ na klientów. Idealnie byłoby odkryć problemy, zanim zostaną zauważone przez klientów. Reaktywne sprawdzanie dzienników zdarzeń pod kątem tego, co się dzieje, niesie ryzyko utraty zaufania ze strony klientów

Kup książkę Poleć książkę

z powodu długiego procesu wykrywania problemów związanych z wydajnością działania i analizy wszelkiego rodzaju aspektów w celu ustalenia przyczyny problemów.

Stosowanie Performance Schema do profilowania wydajności działania MySQL może być bardzo pomocne, co dokładnie wyjaśnimy w rozdziale 3. Ten proces można wykorzystać do wyszukiwania wąskich gardeł, aby egzemplarze mogły robić więcej za pomocą tej samej specyfikacji, co pozwoli zaoszczędzić koszty związane z infrastrukturą lub odpowiedzieć na pytanie „dlaczego to trwa tak długo?". Ze względu na ścisłą integrację z wewnętrznymi komponentami MySQL to narzędzie nie jest przeznaczone wyłącznie do ustalenia, czy są spełnione wymagania dotyczące niezawodności usługi. Do oszacowania poziomu wydajności działania usługi potrzebne jest inne spojrzenie na wydajność działania.

Kilka słów o „testowaniu w produkcji"

Często słyszymy o „testowaniu w produkcji", które wywołuje niesmak u wielu osób. Jednak rzeczywistość jest taka, że testowanie w produkcji ma wiele zalet. Środowisko produkcyjne to miejsce, w którym można zobaczyć, jaki wprowadzone zmiany mają wpływ na pozostałą część systemu, na dużą skalę oraz z rzeczywistym ruchem sieciowym powodowanym przez klienta. Sprawdzić można również wpływ na sąsiednie systemy.

Wykorzystując podstawowe pytanie „czy klienci są zadowoleni?", można sprawdzić kilka kwestii:

- Gdy pętla informacji zwrotnych ze środowiska produkcyjnego jest szybka i ściśle powiązana ze zmianą, wówczas znacznie szybciej można wycofać tę zmianę i dokładnie ją przeanalizować.
- Ta metoda promuje ściślejszą współpracę między zespołami pracującymi nad funkcjonalnością i inżynierami baz danych. Gdy wszystkie strony są zaangażowane w obserwowanie określonych wskaźników i znają wartości, jakie powinny one mieć, wówczas zadanie pomiaru wydajności działania stanie się wysiłkiem grupowym.
- W przypadku regresji wysiłek włożony poza środowiskiem produkcyjnym w celu ustalenia „co się stało?" jest zupełnie inny niż przy odtwarzaniu zestawu testów wydajności emulujących większe ścieżki kodu. Czas, jaki inżynierowie spędzają na debugowaniu rozwiązania, jest znacznie lepiej wykorzystany.

Przechodzimy teraz do kolejnych wskaźników pomocnych w jeszcze dokładniejszym zrozumieniu wrażeń klientów przykładowego sklepu internetowego. Należy zastanowić się nad udostępnianymi przez MySQL wskaźnikami w kategoriach wyniku, a nie konkretnych wartości. Zaprezentujemy również przykłady pokazujące, czego nie można zmierzyć wyłącznie za pomocą wskaźników MySQL.

Monitorowanie dostępności

Sklep internetowy, który czasami przechodzi do trybu offline, może stracić zaufanie klientów. Dlatego tak ważne jest wykorzystywanie dostępności jako oddzielnego wskaźnika oraz jako części procesu sprawdzania wrażeń użytkowników.

Kup książkę Poleć książkę

Dostępność oznacza możliwość reagowania bez błędów na żądania pochodzące od użytkownika. Aby ująć to w ramach standardowego protokołu HTTP, przykładem może być odpowiedź wyraźnie wskazująca sukces (jak udzielenie odpowiedzi z kodem stanu 200) bądź zakończone sukcesem zaakceptowanie żądania razem z obietnicą asynchronicznego zakończenia powiązanych z nim zadań (jak udzielenie odpowiedzi z kodem stanu 202). W czasach monolitycznych systemów w postaci pojedynczego hosta dostępność była zwykłym wskaźnikiem. Obecnie większość architektur ewoluowała w znacznie bardziej zaawansowany sposób odzwierciedlający, jak systemy rozproszone mogą ulegać awarii. Podczas próby zmiany dostępności na wartości wskaźników SLI i SLO architektury bazy danych pod uwagę warto wziąć kilka dalszych szczegółów (pozostajemy przy przykładzie sklepu internetowego):

- Gdy mamy do czynienia z nieuniknioną katastrofą, które funkcje nie podlegają negocjacji, a które zaliczają się do kategorii „dobrze to mieć" (np. czy klient może kontynuować pracę z istniejącym koszykiem na zakupy i sfinalizować transakcję, ale nie może dodawać do niego nowych produktów)?
- Które rodzaje awarii są uznawane za „katastrofalne" (np. brak możliwości wyświetlenia wyników wyszukiwania nie będzie katastrofalny, natomiast brak możliwości sfinalizowania transakcji już tak)?
- Jak wygląda „funkcjonalność zdegradowana" (np. czy można wczytać ogólne rekomendacje zamiast użyć spersonalizowanych na podstawie wcześniejszych zakupów)?
- Jaki jest najkrótszy możliwy czas odzyskiwania systemu po awarii, który można obiecać dla najważniejszych funkcji, biorąc pod uwagę zbiór możliwych scenariuszy awarii (np. jeżeli baza danych zapewniająca obsługę koszyka na zakupy i finalizacji transakcji nie będzie mogła przeprowadzać operacji zapisu, to jak szybko można bezpiecznie przekierować ruch sieciowy do nowego węzła źródłowego?)?

Wybierając zbiór wskaźników reprezentujących dostępność, zespołowi zapewniającemu obsługę klientów trzeba wyjaśnić, że „100% czasu pracy" nie jest rozsądnym celem. Należy skoncentrować się na zapewnieniu klientowi możliwie jak najlepszych wrażeń w świecie, w którym *akceptujemy* to, że awaria komponentu jest nieunikniona.

Preferowana metoda weryfikacji dostępności to klient lub zdalny punkt końcowy. Można to zrobić pasywnie, o ile po stronie klienta jest zapewniony dostęp do dzienników zdarzeń dotyczących połączeń z bazą danych. Innymi słowy, jeżeli aplikacja została utworzona w języku PHP i jest uruchomiona w serwerze Apache, trzeba mieć dostęp do dzienników zdarzeń Apache w celu ustalenia, czy PHP generuje jakiekolwiek błędy podczas nawiązywania połączenia z bazą danych. Dostępność może być sprawdzana również aktywnie. Jeżeli środowisko programisty jest odizolowane i nie ma on dostępu do dzienników zdarzeń klienta, należy rozważyć przygotowanie zdalnego kodu, który będzie przeprowadzał operację w bazie danych i tym samym potwierdzał jej dostępność. To może być naprawdę bardzo proste zapytanie w stylu `SELECT 1`, które weryfikuje, że MySQL otrzymuje zapytania i je przetwarza, ale nie uzyskuje dostępu do warstwy pamięci masowej. Ewentualnie może to być coś znacznie bardziej skomplikowanego, np. odczyt rzeczywistych danych z tabeli bądź wykonanie operacji zapisu i następnie odczytu w celu potwierdzenia poprawności zapisu. Taki rodzaj syntetycznej transakcji z poziomu innego miejsca sieci może pokazać, czy aplikacja jest dostępna.

Kup książkę Poleć książkę

Zdalna weryfikacja dostępności jest użyteczna do śledzenia dostępności celu. Nie pomoże w uzyskaniu dokładniejszych informacji *przed* wystąpieniem problemu. Jeden ze wskaźników MySQL, którego można użyć jako wczesnego sygnału ostrzegawczego w razie problemów związanych z dostępnością, jest licznik stanu `Threads_running` w MySQL. Wskazuje on, ile zapytań jest obecnie wykonywanych w danym hoście bazy danych. Gdy liczba wątków wzrasta bardzo szybko i nie zapowiada się zmiana tej tendencji, może to wskazywać, że wykonywanie zapytań nie będzie się kończyło wystarczająco szybko, a tym samym będą się one kumulować i zużywać zasoby systemu. Umożliwienie wzrostu tego wskaźnika zwykle powoduje, że host bazy danych w pełni wykorzystuje procesor lub pamięć, co może doprowadzić do zamknięcia całego serwera MySQL przez system operacyjny. Jeżeli tak się zdarzy w węźle źródłowym, będzie to stanowiło poważny problem i należy dążyć do dostrzeżenia symptomów tego problemu. Punktem wyjścia może być sprawdzenie liczby rdzeni procesora. Jeżeli wartość `Threads_running` jest większa od liczby rdzeni, może oznaczać, że serwer zbliża się do niebezpiecznego stanu. Ponadto można monitorować zbliżanie do wartości `max_connections`, jako kolejnego punktu danych, który należy sprawdzać pod kątem nadmiernej ilości zadań do wykonania.

W rozdziale 5. znajdziesz więcej informacji o tym, jak zapanować nad niekontrolowanymi wątkami MySQL.

Monitorowanie opóźnienia zapytania

W MySQL wprowadzono wiele długo oczekiwanych usprawnień w zakresie śledzenia czasu potrzebnego na wykonywanie zapytań (*https://dev.mysql.com/doc/refman/8.0/en/performance-schema-statement-summary-tables.html*). Stos monitorowania należy zdecydowanie wykorzystać do śledzenia tych trendów podczas modyfikowania kodu aplikacji. Jednak mimo to nadal nie otrzymujemy pełnego obrazu wrażeń klienta, zwłaszcza biorąc pod uwagę to, jak projektowana jest nowoczesna architektura oprogramowania. Poza wewnętrznym śledzeniem opóźnienia trzeba również spojrzeć na opóźnienie z perspektywy aplikacji i tego, co się dzieje, gdy to opóźnienie wzrasta. Dlatego oprócz bezpośredniego śledzenia w bazie danych opóźnienia podczas wykonywania zapytań potrzebne są również narzędzia klientów dostarczające informacje o czasie potrzebnym na wykonanie zapytania, aby w ten sposób zebrać jak najwłaściwsze dane o wrażeniach użytkownika. Streszczenie tych wszystkich przykładowych wskaźników od klientów (zwłaszcza gdy infrastruktura jest rozbudowywana) można przeprowadzić za pomocą płatnych narzędzi, takich jak Datadog lub SolarWinds Database Performance Monitor, albo za pomocą narzędzi open source, takich jak PMM. Istnieje pewien obszar, na którym najważniejsza będzie ścisła współpraca z programistami aplikacji. Trzeba wiedzieć, jak zespoły aplikacji dokonują pomiaru z perspektywy aplikacji, i nieco dokładniej zapoznać się z elementami odstającymi, używając do tego narzędzi typu Honeycomb i Lightstep.

Monitorowanie pod kątem błędów

Czy zachodzi potrzeba śledzenia każdego błędu i informowania o jego wystąpieniu? To zależy.

Samo istnienie błędów klienta MySQL podczas działania usługi nie wskazuje na awarię. W świecie systemów rozproszonych mamy scenariusze, w których klienty mogą napotykać nieuniknione błędy.

Kup książkę Poleć książkę

W wielu przypadkach zostaną one usunięte przez zwykłe ponowne wykonanie zapytania. Natomiast *współczynnik* występujących błędów we flocie usług obsługujących zapytania bazy danych w infrastrukturze może wskazywać na nadchodzące problemy. Oto kilka przykładów błędów po stronie klienta, które zwykle mogą pozostać niezauważone, a wzrost ich liczby będzie oznaczał zbliżające się problemy.

Brak przekroczenia czasu oczekiwania przy oczekiwaniu na blokadę

Jeżeli klient będzie zgłaszał gwałtowny wzrost liczby wystąpień tego błędu, może to wskazywać na ogromny skok w węźle źródłowym rywalizacji podczas nakładania blokad na rekordy. W efekcie transakcje są powtarzane i mimo to kończą się niepowodzeniem. Taka sytuacja może być zapowiedzią przestoju w zakresie operacji zapisu.

Zerwane połączenia

Jeżeli klient zgłasza nagły wzrost liczby przerwanych połączeń, może to być zapowiedzią problemów na dowolnej warstwie dostępu istniejącej między klientami i egzemplarzami baz danych. Jeżeli liczba tych błędów nie będzie monitorowana, skutkiem może być ogromna liczba ponownych prób po stronie klienta, co przekłada się na większe zużycie zasobów.

Istnieje możliwość zdefiniowania zbioru zmiennych serwerowych `Connection_errors_xxx` (*https://dev.mysql.com/doc/refman/8.0/en/server-status-variables.html#statvar_Connection_errors_xxx*), gdzie `xxx` oznacza odmienne rodzaje błędów połączenia. Nagły wzrost wartości tych liczników może być sygnałem wskazującym, że coś nowego i nietypowego obecnie nie działa.

Czy mamy błędy, których pojedyncze wystąpienie oznacza problem i dlatego trzeba się takim błędem zająć? Tak.

Na przykład błąd informujący o działaniu egzemplarza MySQL w trybie tylko do odczytu jest oznaką problemów, nawet jeśli taki błąd nie pojawia się zbyt często. To może oznaczać istnienie repliki promowanej do roli źródła, a mimo to wciąż działającej w trybie tylko do odczytu (Twoje repliki są uruchomione w trybie tylko do odczytu, prawda?), co będzie dla klastra oznaczało czas przestoju operacji zapisu. Ewentualnie może oznaczać problem na warstwie dostępu przekazującej do repliki ruch sieciowy w postaci operacji zapisu. W każdym razie to nie jest sygnał błędu rozwiązywanego po ponownym wykonaniu zapytania.

Innym przykładem błędu sygnalizującego poważny problem jest „zbyt wiele połączeń” lub pojawiający się na poziomie systemu operacyjnego „brak możliwości utworzenia nowego wątku”. To wskazuje na utworzenie warstwy aplikacji i pozostawienie większej liczby otwartych połączeń niż dozwolona w serwerze bazy danych, np. z powodu ustawienia zmiennej serwerowej `max_connections` lub ze względu na liczbę wątków, które może otworzyć proces MySQL. Te błędy przekładają się natychmiast na błędy o kodzie 5xx w aplikacji i w zależności od projektu aplikacji również mogą mieć wpływ na klientów.

Widać wyraźnie, że pomiar wydajności działania i wybór błędów uwzględnianych w SQL to problem bardziej związany z komunikacją niż względami technicznymi, więc trzeba być na to przygotowanym.

Kup książkę Poleć książkę

Monitorowanie proaktywne

Jak wcześniej wspomnieliśmy, monitorowanie wskaźnika SLO koncentruje się na tym, czy klienci są zadowoleni. To pomaga skupić się na poprawie wrażeń klientów, gdy są niezadowoleni, oraz na innych zadaniach, takich jak zmniejszenie ilości pracy, gdy klienci są zadowoleni. W takim ujęciu został pominięty jeden kluczowy obszar: monitorowanie proaktywne.

Jeżeli powrócimy do przykładu sklepu internetowego i planowanego sposobu monitorowania wrażeń klientów, to można to rozwinąć jeszcze bardziej. Załóżmy, że nie ma poważniejszych awarii komponentów, a jednocześnie zwiększa się liczba skarg klientów narzekających na „powolność" działania lub okazyjne błędy, które same znikają. W jaki sposób można wyśledzić takie zachowanie? To może być bardzo trudne zadanie, jeżeli nie została zdefiniowana bazowa wydajność działania. Panele z informacjami i skrypty wywołujące powiadomienia na żądanie można określić jako *stan ciągłego monitorowania*. Dzięki temu monitorowaniu wiadomo o czymś nieprzewidzianym w systemie, niezależnie od tego, czy została w nim wprowadzona zmiana. Istnieją ważne narzędzia informujące o zbliżających się problemach, zanim ujawnią się one użytkownikom.

Równowaga konieczna do osiągnięcia za pomocą monitorowania zawsze powinna być możliwa do osiągnięcia i jednocześnie pozostawać prawdziwym wskaźnikiem. Informowanie o zajęciu w 100% pamięci masowej przeznaczonej dla bazy danych jest za późne, ponieważ w tym momencie usługa już nie działa. Z kolei powiadomienie o zajęciu pamięci masowej na poziomie 80% może być zbyt wolne lub niewykonalne, jeśli ilość danych nie przyrasta bardzo szybko.

Warto wspomnieć o użytecznych sygnałach, które można monitorować i które nie mają bezpośredniego wpływu na klienta.

Wzrost poziomu użycia pamięci masowej

Śledzenie wzrostu poziomu użycia pamięci masowej może nie być interesującym Cię wskaźnikiem, dopóki brak wolnego miejsca nie stanie się problemem. Wtedy rozwiązanie może wymagać czasu i będzie wpływać na produkt. Znacznie lepiej jest poznać sposoby śledzenia tego wskaźnika i przygotować plan pozwalający na zminimalizowanie ryzyka, a także ustalenie bezpiecznych poziomów progowych, o których przekroczeniu system będzie powiadamiał.

Istnieje wiele różnych strategii, które można wykorzystać podczas monitorowania wzrostu poziomu użycia pamięci masowej. W tym podpunkcie omówimy je w kolejności od idealnej do absolutnego minimum.

Jeżeli narzędzie monitorowania na to pozwala, śledzenie wzrostu poziomu użycia pamięci masowej może okazać się niezwykle użyteczne. Zawsze istnieją scenariusze, w których dostępna ilość miejsca na dysku szybko się zmniejsza, co stanowi niebezpieczeństwo dla zapewnienia dostępności usługi. Operacje takie jak długo wykonywane transakcje z ogromnymi dziennikami przywracania lub modyfikacje tabel to przykłady pokazujące, dlaczego dysk może zostać zbyt szybko zapełniony. Istnieje wiele historii, w których nadmierna ilość rejestrowanych danych bądź zmiana wzorca wstawiania danych pozostały niezauważone aż do chwili, gdy w „bazie danych" zabrakło miejsca. Dopiero wtedy zostały wygenerowane różnego rodzaju powiadomienia.

Kup książkę Poleć książkę

Jeżeli śledzenie wzrostu poziomu użycia pamięci masowej jest niemożliwe (nie wszystkie narzędzia monitorowania oferują taką funkcjonalność), można zdefiniować kilka mniejszych wartości progowych z ostrzeżeniami, które będą generowane tylko w godzinach pracy, oraz wyższych wartości progowych z ostrzeżeniami, które będą generowane tylko poza godzinami pracy. Dzięki temu zespół będzie miał rękę na pulsie w godzinach pracy, poza nimi zaś alarm będzie wszczynany tylko wtedy, kiedy sytuacja naprawdę będzie tego wymagała.

Jeżeli nie można monitorować stopnia wzrostu poziomu użycia pamięci masowej ani zdefiniować wielu wartości progowych dla tego samego wskaźnika, trzeba przynajmniej zdefiniować jedną wartość progową dotyczącą pamięci masowej, o której przekroczeniu będzie informowany dyżurny inżynier. Ta wartość musi być na tyle mała, aby zapewnić czas potrzebny na podejmowanie działań i zwolnienie miejsca na dysku, gdy zespół będzie analizował powód alarmu i szukał sposobów na trwałe usunięcie tego powodu. Warto sprawdzić maksymalną przepustowość zapisu dysku (wyrażoną w MB/s) i wykorzystać tę informację do ustalenia, ile czasu zajmie zapełnienie dysku przy wykonywaniu operacji zapisu z maksymalną przepustowością. Taka ilość czasu jest wcześniej potrzebna, aby można było uniknąć zdarzenia.

W rozdziale 4. omówimy temat systemu operacyjnego i konfiguracji sprzętowej, które mają związek ze sposobem używania pamięci masowej przez MySQL, oraz kompromisy, jakie trzeba uwzględnić przy podejmowaniu decyzji dotyczących wzrostu poziomu użycia pamięci masowej. Trzeba się spodziewać, że w pewnym momencie (mamy nadzieję) Twój biznes rozrośnie się na tyle, że wszystkich danych nie będzie się dało przechowywać w jednym klastrze serwerów. Nawet jeśli używasz środowiska chmury, które pozwala rozszerzać woluminy, i tak trzeba to zaplanować. Dlatego zawsze należy mieć zdefiniowaną wartość progową wolnej ilości miejsca na dysku, co zapewni czas potrzebny na planowanie i przeprowadzenie niezbędnej ekspansji bez paniki.

Ogólnie rzecz biorąc, trzeba zapewnić monitorowanie pod kątem wzrostu poziomu użycia pamięci masowej, nawet jeśli wydaje się, że to wczesny etap rozwoju usługi i jeszcze jest na to za wcześnie. To jest jeden z tych obszarów wzrostu, który zaskakuje praktycznie wszystkich nieprzygotowanych.

Wzrost liczby połączeń

Wraz z rozwojem przedsięwzięcia warstwą najczęściej wzrastającą liniowo jest warstwa aplikacji. Konieczne jest zapewnienie większej liczby egzemplarzy potrzebnych do obsługi logowania, koszyków na zakupy, przetwarzania żądań bądź wykonywania innych zadań w kontekście produktu. Wszystkie te dodane egzemplarze rozpoczynają nawiązywanie kolejnych połączeń z hostem bazy danych. Ten wzrost można przynajmniej częściowo złagodzić przez dodanie replik, użycie replikacji jako miary skalowania, a nawet zastosowanie warstw pośrednich, takich jak ProxySQL, w celu odizolowania wzrostu zainteresowania frontendem od powodowanego przez połączenia obciążenia bezpośrednio w bazie danych.

Podczas gdy ruch sieciowy wzrasta, serwer bazy danych może obsługiwać tylko określonej wielkości pulę połączeń, która to wielkość jest skonfigurowana za pomocą zmiennej `max_connections`. Gdy całkowita liczba połączeń z serwerem osiągnie maksimum, baza danych nie będzie zezwalała na nawiązywanie nowych, co często przyczynia się do problemów polegających na braku możliwości nawiązywania nowych połączeń z bazą danych. To często prowadzi do zwiększenia liczby błędów wyświetlanych użytkownikom.

Kup książkę Poleć książkę

Monitorowanie wzrostu liczby połączeń ma zagwarantować, że zasoby nie zostaną wyczerpane do punktu, w którym to już zagraża dostępności bazy danych. Takie niebezpieczeństwo może objawić się na dwa różne sposoby:

- Warstwa aplikacji otwiera wiele połączeń, których nie używa. To prowadzi do ryzyka nieuzasadnionego wykorzystania całej puli połączeń. Wyraźnym sygnałem takiego stanu rzeczy jest zwiększenie liczby połączeń (`threads_connected`) i niska wartość `threads_running`.
- Warstwa aplikacji aktywnie używa wielu połączeń, co prowadzi do niebezpieczeństwa przeciążenia bazy danych. Tę sytuację można odróżnić od poprzedniej, obserwując duże (setki, tysiące) wartości zmiennych `threads_connected` *oraz* `threads_running`, które ciągle się zwiększają.

Podczas konfigurowania opcji monitorowania pod względem liczby połączeń dobrze jest bazować na wartościach procentowych, a nie na liczbach bezwzględnych. Wartość procentowa `threads_connected/max_connections` pokazuje, jak wzrost licznika połączeń węzła aplikacji zbliża się do maksymalnej wielkości puli połączeń dozwolonej przez bazę danych. To pomaga w monitorowaniu pod kątem pierwszego stanu wskazującego na pojawiający się problem ze wzrostem liczby połączeń.

Oddzielnie można śledzić poziom zajętości hosta bazy danych i alarmować w razie potrzeby, co znajduje odzwierciedlenie w wartości `threads_running`, jak to wyjaśniliśmy już wcześniej. Jeżeli ta wartość przekracza 100 wątków, zaczyna być widoczny nadmierny poziom użycia procesora i pamięci, co zasadniczo jest ogólnym sygnałem wskazującym na wysokie obciążenie hosta bazy danych. To stanowi powód do obaw o dostępność bazy danych, ponieważ problem może się powiększyć do tego stopnia, że proces MySQL zostanie zamknięty przez system operacyjny. Często stosowanym szybkim rozwiązaniem jest zamykanie innych procesów lub wykorzystanie narzędzia, np. *pt-kill*, automatyzującego tę operację, w celu zmniejszenia obciążenia. Następnie trzeba ustalić, dlaczego baza danych znalazła się w takim stanie — do tego można wykorzystać analizę zapytania, jak to już wcześniej omówiliśmy.

Burza połączeń to sytuacja w systemie produkcyjnym, gdy na warstwie aplikacji obserwuje się wzrastający poziom opóźnienia i wydłużenie czasu udzielenia odpowiedzi wraz z nawiązywaniem kolejnych połączeń z warstwą bazy danych. Skutkiem może być znacznie obciążenie bazy danych, ponieważ będzie musiała obsługiwać ogromną ilość nowych połączeń, co z kolei będzie wykorzystywało zasoby potrzebne do wykonywania zapytań. Burza połączeń może spowodować nagły i gwałtowny spadek liczby dostępnych połączeń w `max_connections` i zwiększyć zagrożenie dla dostępności bazy danych.

Opóźnienie replikacji

MySQL ma natywną funkcjonalność replikacji, która powoduje przekazywanie danych z jednego serwera, *źródła*, do jednego lub więcej innych serwerów, nazywanych *replikami*. Opóźnienie między zapisem danych w źródle i dostępnością tych danych w replikach jest określane mianem *opóźnienia replikacji*. Jeżeli aplikacja musi odczytywać dane z replik, to opóźnienie może spowodować, że odczytane dane będą niespójne, ponieważ repliki mogły jeszcze nie zastosować wszystkich zmian. W przykładzie serwisu społecznościowego użytkownik może skomentować post

Kup książkę Poleć książkę

opublikowany przez innego użytkownika. Te dane zostaną zapisane w źródle, a następnie przekazane do replik. Gdy użytkownik spróbuje wyświetlić odpowiedź, to jeśli aplikacja wykona żądanie do serwera opóźnionego, replika może jeszcze nie mieć żądanych danych. W efekcie użytkownik będzie zdezorientowany i może pomyśleć, że komentarz nie został zapisany. Strategie radzenia sobie z opóźnieniem replikacji omówiliśmy dokładnie w rozdziale 9.

Opóźnienie to jeden z tych wskaźników, które mogą być wartością SLI powodującą wywoływanie incydentów. To również długoterminowy trend wskazujący na potrzebę większej zmiany architekturalnej. W dłuższej perspektywie nawet jeśli nigdy nie pojawi się opóźnienie replikacji wpływające na wrażenia użytkownika usługi, wciąż jest to sygnał, przynajmniej pośredni, że liczba operacji zapisu ze źródła jest większa, niż repliki mogą obsłużyć przy ich obecnej konfiguracji. Tę sytuację można porównać do kanarka w kopalni węgla: odpowiednio wczesna reakcja na jego zachowanie może uchronić przed ogromnym problemem.

Należy zachować ostrożność podczas powiadamiania o opóźnieniu replikacji. Natychmiastowa reakcja nie zawsze może być możliwa. Podobnie jeśli dane nie są odczytywane z replik, dobrze jest rozważyć, jak agresywnie system monitorowania informuje o takiej sytuacji. Powiadomienia, zwłaszcza poza godzinami pracy, powinny pozwalać na podejmowanie pewnych działań.

Opóźnienie replikacji to jeden z tych wskaźników, które mogą wpływać na decyzje natychmiastowe i taktyczne. Ponadto obserwowanie trendu długoterminowego pomaga uniknąć poważniejszych problemów i pozwala z wyprzedzeniem zareagować na krzywą wzrostu opóźnienia replikacji.

Poziom wykorzystania operacji wejścia-wyjścia

Jednym z nigdy nie kończących się wyzwań inżyniera bazy danych jest „wykonać jak najwięcej zadań na danych w pamięci, ponieważ tak jest szybciej". Wprawdzie to niewątpliwie jest właściwe podejście, ale jednocześnie trudne do zastosowania w 100% przypadków, ponieważ to będzie oznaczało, że dane całkowicie mieszczą się w pamięci. W takim przypadku „skalowanie" nie jest zadaniem, na którego wykonywanie należy tracić energię.

Gdy infrastruktura bazy danych jest skalowana i dane nie mieszczą się już w pamięci, trzeba wiedzieć, że kolejnym najlepszym podejściem jest nieodczytywanie ogromnej ilości danych z dysku, w którym zapytania utknęły, oczekując na swoją kolej uzyskania dostępu do cennych cykli operacji wejścia-wyjścia. To się nie zmieniło nawet w erze, w której niemal wszystkie napędy w serwerach baz danych to SSD. Gdy wielkość danych wzrośnie, a zapytania muszą skanować więcej danych w celu spełnienia zapytań, oczekiwanie na operacje wejścia-wyjścia staje się wąskim gardłem dla wzrostu ilości ruchu sieciowego.

Monitorowanie aktywności dyskowych operacji wejścia-wyjścia pomaga uniknąć degradacji wydajności działania, zanim stanie się ona oczywista dla klienta. Istnieje kilka aspektów, które można monitorować i osiągnąć zamierzony cel. Narzędzia takie jak *iostat* mogą pomóc w monitorowaniu pod kątem oczekiwania na dyskowe operacje wejścia-wyjścia. Należy monitorować i powiadamiać, gdy serwer bazy danych ma wiele wątków w stanie `IOwait` — to wyraźny sygnał, że znajdują się one w kolejce oczekującej na dostępność zasobów dyskowych. Można to sprawdzić

Kup książkę Poleć książkę

przez monitorowanie `IOutil` przez pewien czas, np. dzień, dwa lub nawet tydzień. Wskaźnik `IOutil` to wartość procentowa określająca ogólną pojemność dostępu dysku systemu. Jeżeli ta wartość będzie bliska 100% przez pewien czas, gdy nie jest tworzona kopia zapasowa, może to wskazywać na przeprowadzanie pełnego skanowania tabeli i nieefektywne zapytania. Ponadto należy monitorować ogólny poziom użycia pojemności dyskowych operacji wejścia-wyjścia jako wartości procentowej, ponieważ może to ostrzec przed sytuacją, w której dostęp do dysku stanie się wąskim gardłem dla wydajności działania bazy danych.

Przestrzeń automatycznej inkrementacji

Jedną z mniej znanych pułapek podczas używania MySQL jest automatyczna inkrementacja kluczy podstawowych, które domyślnie są tworzone jako liczby całkowite ze znakiem i dostępna dla nich przestrzeń może się skończyć. Tak się stanie po wykonaniu takiej liczby operacji wstawiania, że automatycznie inkrementowany klucz osiągnie wartość maksymalną dla używanego typu danych. Podczas planowania wskaźników monitorowanych w dłuższej perspektywie trzeba pamiętać o monitorowaniu również przestrzeni liczb całkowitych we wszystkich tabelach, w których klucz podstawowy jest automatycznie inkrementowany. Jest to prosta operacja, która na pewno pozwoli uniknąć poważnego problemu w przyszłości, ponieważ można wcześniej przewidzieć potrzebę użycia większej przestrzeni dla klucza.

W jaki sposób można monitorować przestrzeń dla kluczy? Do dyspozycji jest kilka rozwiązań. Jeżeli jest już używane oprogramowanie PMM i jego komponent eksporter dla oprogramowania Prometheus, wszystko jest przygotowane i wystarczy użyć opcji `-collect.auto_increment.columns`. Natomiast jeśli zespół nie używa oprogramowania Prometheus, można skorzystać z przedstawionego tutaj zapytania, które można zmodyfikować do postaci generatora wskaźnika lub powiadomienia, gdy tabele będą się zbliżały do końca przestrzeni dla kluczy (*https://vettabase.com/blog/monitor-auto-increment-usage/*). To zapytanie opiera się na tabelach `information_schema`, które zawierają wszystkie metadane dotyczące tabel znajdujących się w egzemplarzu bazy danych.

```
SELECT
    t.TABLE_SCHEMA AS `schema`,
    t.TABLE_NAME AS `table`,
    t.AUTO_INCREMENT AS `auto_increment`,
    c.DATA_TYPE AS `pk_type`,
    (
        t.AUTO_INCREMENT /
        (CASE DATA_TYPE
            WHEN 'tinyint'
                THEN IF(COLUMN_TYPE LIKE '%unsigned',
                    255,
                    127
                )
            WHEN 'smallint'
                THEN IF(COLUMN_TYPE LIKE '%unsigned',
                    65535,
                    32767
                )
            WHEN 'mediumint'
                THEN IF(COLUMN_TYPE LIKE '%unsigned',
                    16777215,
```


Kup książkę Poleć książkę

```
                    8388607
                )
            WHEN 'int'
                THEN IF(COLUMN_TYPE LIKE '%unsigned',
                    4294967295,
                    2147483647
                )
            WHEN 'bigint'
                THEN IF(COLUMN_TYPE LIKE '%unsigned',
                    18446744073709551615,
                    9223372036854775807
                )
        END / 100)
    ) AS `max_value`
    FROM information_schema.TABLES t
    INNER JOIN information_schema.COLUMNS c
        ON t.TABLE_SCHEMA = c.TABLE_SCHEMA
        AND t.TABLE_NAME = c.TABLE_NAME
    WHERE
        t.AUTO_INCREMENT IS NOT NULL
        AND c.COLUMN_KEY = 'PRI'
        AND c.DATA_TYPE LIKE '%int'
;
```

Wybór klucza podstawowego wiąże się z wieloma niuansami i kontekstem, ogólnie i podczas zarządzania automatyczną inkrementacją, co dokładniej omówimy w rozdziale 6.

Tworzenie kopii zapasowej i czas przywracania z niej danych

Planowanie długoterminowe dotyczy nie tylko wzrostu, gdy wszystko działa normalnie, ale również odzyskiwania danych w akceptowalnych ramach czasowych. W rozdziale 10. znajdziesz dokładne omówienie zagadnień związanych z odzyskiwaniem danych, a w rozdziale 13. wyjaśnimy te zagadnienia jako część procesu kontroli zapewnienia zgodności. Natomiast w tym miejscu chcemy przypomnieć, że dobry plan ratunkowy działa tylko wtedy, gdy zostanie przeanalizowany, a jego cele będą odpowiednio dostosowane.

Sharding funkcjonalny i poziomy

W tym rozdziale oraz w innych miejscach książki wymienialiśmy sharding lub partycjonowanie jako różne sposoby podziału danych na oddzielne egzemplarze w celu przeprowadzenia skalowania. Chcemy w tym miejscu wyjaśnić, co mamy na myśli i czym te pojęcia się różnią. Dzięki temu unikniesz dezorientacji podczas lektury pozostałej części książki.

Sharding funkcjonalny oznacza podział określonych tabel służących konkretnym funkcjom biznesowym na dedykowany klaster w celu oddzielnego zarządzania tym zbiorem danych — jego czasem nieprzerwanego działania, wydajnością, a nawet kontrolą dostępu.

Sharding poziomy zachodzi, gdy wielkość zbioru danych przekroczyła tę, którą można niezawodnie obsłużyć za pomocą pojedynczego klastra. W takiej sytuacji dane są dzielone na wiele klastrów i udostępniane przez wiele węzłów, bazując przy tym na pewnym mechanizmie umożliwiającym znalezienie niezbędnego podzbioru.

Kup książkę Poleć książkę

Jeżeli baza danych osiągnęła wielkość oznaczającą, że przywrócenie z kopii zapasowej funkcjonalności biznesowej o znaczeniu krytycznym będzie trwało dłużej, niż jest to akceptowane, wówczas jeśli wszystko inne działa świetnie, trzeba będzie zmodyfikować cel MTTR, zmienić definicję „funkcjonalności biznesowej o znaczeniu krytycznym" lub znaleźć sposób pozwalający skrócić czas przywracania z kopii zapasowej. Oto kilka kwestii, które należy rozważyć podczas planowania operacji odzyskiwania danych po awarii:

- Trzeba *bardzo dokładnie* określać funkcjonalność wchodzącą w zakres danego celu odzyskiwania danych i jeśli zachodzi potrzeba, ustalić, czy dane związane z tą funkcjonalnością faktycznie muszą znajdować się w oddzielnym klastrze, aby oczekiwania pozostały realne do spełnienia.
- Jeżeli niemożliwe jest partycjonowanie funkcjonalne danych na wiele mniejszych egzemplarzy, to cały zbiór danych nie zmieści się w danym celu odzyskiwania danych za pomocą kopii zapasowej. Zbiór danych, którego przywrócenie z kopii zapasowej będzie trwało najdłużej, wpływa na czas trwania procesu odzyskiwania danych.
- Trzeba się upewnić o posiadaniu zautomatyzowanych metod testowania (wybrane przykłady zostaną omówione w rozdziale 10.). Należy sprawdzić, ile czasu zajmie przywrócenie kopii zapasowej z pliku do działającej bazy danych, w której za pomocą replikacji zostaną wprowadzone wszystkie zmiany od chwili wykonania kopii zapasowej. Te informacje należy przechowywać przez wystarczająco długi czas, aby poznać trend długoterminowy (przynajmniej rok). To jest jeden ze wskaźników, których wartość może się wydłużyć i stać zaskakująco duża, jeśli monitorowanie nie będzie zautomatyzowane.

Zobaczysz, że w wielu przykładach wskaźników długoterminowych, które wkrótce omówimy, nie zawsze jest wskazywana konieczność przeprowadzenia shardingu danych, funkcjonalnego lub poziomego. Celem jest wyraźne podkreślenie faktu, że jeśli rozważany jest sharding w przypadku pojawiania się incydentów związanych przede wszystkim z pojemnością, wówczas takie rozwiązanie jest brane pod uwagę zdecydowanie zbyt późno. Podział danych na łatwiejsze w zarządzaniu fragmenty nie rozpoczyna się w chwili, gdy te dane stały się zbyt duże dla pojedynczego klastra, lecz znacznie wcześniej, w trakcie określania celów, aby zapewnić klientowi jak najlepsze wrażenia podczas używania produktu bądź usługi.

Ustalenie czasu potrzebnego do przywrócenia danych może pomóc w określeniu oczekiwań względem tego, co należy zrobić w przypadku rzeczywistej katastrofy. To może również ujawnić, że czas operacji odzyskiwania danych jest dłuższy niż oczekiwany, co może wskazywać na konieczność shardingu danych.

Pomiar długoterminowej wydajności działania

Wybór SLI i SLO dla codziennych operacji to dopiero początek. Trzeba rozumieć problem w szerszym ujęciu i skoncentrować się na określonych wskaźnikach hosta zamiast analizować ogólną wydajność działania systemu i wrażenia użytkownika. W tym podrozdziale przedstawimy strategie, które można wykorzystać podczas planowania długoterminowego stanu systemu.

Kup książkę Poleć książkę

Poznanie rytmu pracy firmy

Konieczne jest ustalenie rytmu pracy firmy, ponieważ to zawsze będzie czas, w którym wszystkie wartości SLO zostaną najdokładniej przetestowane i przeanalizowane przez najważniejszych klientów. Analiza może ujawnić, że ruch sieciowy w godzinach szczytu jest o rząd wielkości większy niż „przeciętnie" i będzie to miało duże konsekwencje, gdy infrastruktura bazy danych będzie nieprzygotowana. W kontekście infrastruktury bazy danych może to mieć przełożenie na znacznie większą liczbę zapytań do spełnienia, znacznie większe obciążenie serwerów aplikacji, a także większe straty finansowe w przypadku awarii dotykającej operacje zapisu. Oto kilka przykładów pomocnych w zrozumieniu cyklu biznesowego firmy:

Sklep internetowy

W wielu krajach od połowy listopada do końca roku trwa gorący sezon, w którym sprzedaż jest znacznie większa. To oznacza wiele dodatkowych koszyków na zakupy, znacznie więcej jednoczesnych operacji sprzedaży i dużo większe straty w przypadku wystąpienia awarii w tym okresie.

Usługi dla ludności

W USA listopad to miesiąc, w którym wielu pracowników dokonuje wyboru ubezpieczenia zdrowotnego w trakcie tzw. „wolnego zapisu", co może oznaczać znacznie większy ruch sieciowy.

Internetowa kwiaciarnia

Walentynki to dla firmy zajmującej się dostawą kwiatów najbardziej pracowity dzień w roku, gdyż liczba zamówień jest wtedy rekordowo duża.

Jak można zobaczyć, te cykle różnią się znacznie między sobą, w zależności od spełnianych przez daną firmę potrzeb klientów. Trzeba koniecznie poznać cykl firmy oraz jego wpływ na przychody i wizerunek firmy. To oznacza konieczność przygotowania się na sprostanie wymaganiom bez wpływu na stabilność działających systemów.

W zakresie pomiaru wydajności działania infrastruktury bazy danych używanej w firmie bardzo ważne jest, aby pomiaru nie dokonywać w oderwaniu od innych ważnych wskaźników monitorowanych przez inżynierów organizacji. Wydajność działania bazy danych powinna być częścią większej konwersacji dotyczącej wydajności całego stosu, a nie być traktowana jako przypadek specjalny. Należy rozpocząć od użycia tych samych narzędzi, z których korzystają pozostali inżynierowie organizacji. Wskaźniki i panele stosowane do określenia wydajności działania warstwy bazy danych powinny być równomiernie dostępne jako wskaźniki warstwy aplikacji lub nawet w tych samych panelach. Takie nastawienie, niezależnie od technologii lub producenta rozwiązania, pozwoli stworzyć środowisko, w którym każdy będzie zaangażowany w wydajność działania pełnego stosu. To spowoduje także zmniejszenie barier, jakie mogą odczuwać inżynierowie między tworzoną funkcjonalnością i obsługującą ją bazą danych.

Kup książkę Poleć książkę

Efektywne śledzenie wskaźników

Jest wiele kwestii, które trzeba wziąć pod uwagę podczas przygotowywania długoterminowych planów dotyczących firmy. Oto wybrane z nich:

- planowanie przyszłej pojemności;
- przewidywanie, kiedy konieczne będzie wprowadzenie poważnych usprawnień, a kiedy wystarczą jedynie drobne zmiany;
- przygotowanie się na większe koszty działania infrastruktury.

Trzeba zapewnić nie tylko możliwość monitorowania stanu infrastruktury magazynu danych w określonej chwili, ale również trendu poprawy lub degradacji wydajności działania na dłuższą metę. To oznacza nie tylko definiowanie wartości SLI i SLO, ale także ustalenie, które SLI i SLO pozostają cennymi wskaźnikami dla długoterminowych trendów. Prawdopodobnie okaże się, że nie wszystkie wskaźniki używane do podejmowania decyzji na krótką metę nadają się również do planowania działalności firmy w dłuższym czasie.

Zanim przejdziemy do wyjaśnienia, które wskaźniki są ważne w planowaniu długoterminowym, najpierw powiemy nieco o narzędziach pomocnych w monitorowaniu długoterminowych trendów.

Stosowanie narzędzi monitorowania do analizy wydajności działania

Sprawdzanie wydajności działania jest ważne zarówno w sytuacji „mamy problem", jak i podczas śledzenia trendu na dłuższą metę. Wybór narzędzia przechowującego interesujące Cię wskaźniki jest tak samo ważny jak same wskaźniki. Jaki jest sens wyboru dobrej wartości SLI, gdy trendu na przestrzeni czasu nie będzie można zobaczyć w sposób powiązany z pozostałymi wskaźnikami stosowanymi w organizacji?

Dziedzina narzędzi monitorowania przeżywa prawdziwy rozkwit i istnieje wiele świetnych rozwiązań, spośród których można wybierać. Celem jest większa przejrzystość i koncentracja na wynikach, a nie konkretnych wartościach. W celu zapewnienia sukcesu stosowi infrastruktury osiągnięcie sukcesu w trakcie monitorowania jest zadaniem dla całego zespołu.

Zamiast wymieniać tutaj konkretne narzędzia przedstawimy listę wybranych ważnych funkcji i aspektów, które trzeba wziąć pod uwagę podczas ustalania, czy dane narzędzie jest dobre do śledzenia trendu długookresowego.

Powiedz „nie" wartościom średnim

Niezależnie od tego, czy samodzielnie zarządzasz wskaźnikami w organizacji, czy korzystasz z rozwiązania typu SaaS (ang. *software as a service*), trzeba zachować ostrożność pod względem tego, jak rozwiązanie normalizuje dane w celu ich długotrwałego przechowywania. Wiele rozwiązań domyślnie agreguje dane długoterminowe na postać wartości średnich (przykładem może być Graphite) i to stanowi duży problem. Jeżeli trzeba przyjrzeć się trendowi wskaźnika na przestrzeni czasu dłuższego niż kilka tygodni, średnia *ukryje* wartości odstające. Dlatego jeśli celem będzie ustalenie, czy liczba dyskowych operacji wejścia-wyjścia podwoi się w następnym roku,

Kup książkę Poleć książkę

wykres przedstawiający wartości średnie prawdopodobnie dostarczy niepoprawne dane zapewniające mylne poczucie bezpieczeństwa. Podczas analizowania danych miesięcznych zawsze należy przyglądać się wartościom odstającym, aby dostrzegać sporadyczne skoki.

Percentyl to Twój przyjaciel

Percentyl opiera się na kolejności punktów danych w określonym czasie i usuwa najwyższe wartości, w zależności od percentyla docelowego (np. jeśli chcesz otrzymać 95 percentyl, usuń 5% najwyższych wartości). To doskonały sposób, dzięki któremu szukane dane będą wizualnie podobne do wartości SLI i SLO. Jeżeli można utworzyć wykres pokazujący 95. percentyl czasu udzielenia odpowiedzi, będzie go można znacznie łatwiej dopasować do docelowej wartości SLO dla ukończonych żądań aplikacji. Ponadto wskaźniki bazy danych będą miały sens na przykład dla pracowników działu obsługi klienta i inżynierów, a nie tylko dla zespołu odpowiedzialnego za obsługę bazy danych.

Długi czas przechowywania wskaźników i wydajność działania

To może wydawać się oczywiste, ale duże znaczenie ma również wydajność działania narzędzia przeznaczonego do monitorowania dłuższych okresów. Jeżeli analizowane są rozwiązania do obsługi trendów biznesowych, trzeba testować, jak zmienia się sposób działania narzędzia, gdy dane dotyczą coraz dłuższych okresów. Rozwiązanie dotyczące wskaźników jest dobre tylko wtedy, kiedy zapewnia dostępność danych, a nie jedynie szybkość ich przetwarzania lub długi czas ich przechowywania.

W tym punkcie wymieniliśmy wybrane aspekty narzędzia przeznaczonego do monitorowania w dłuższej perspektywie czasu. Warto teraz dowiedzieć się, jak przedstawione dotychczas informacje dotyczące wyboru SLI i SLO mogą pomóc w przygotowaniu architektury dla posiadanych danych.

Stosowanie SLO do przygotowania architektury danych

Zapewnienie spójnych i dobrych wrażeń klienta podczas rozbudowy firmy nie jest małym wysiłkiem. Gdy firma się rozrasta, zachowanie tego samego SLO, nie wspominając już o ustaleniu bardziej ambitnych celów, staje się coraz trudniejsze. Weźmy na przykład dostępność — każdy chce osiągnąć wyrażony jak największą liczbą dziewiątek czas nieprzerwanej pracy dla operacji zarówno odczytu, jak i zapisu. Jednak im bardziej ambitny cel SLO, tym cięższa praca wymagana do jego osiągnięcia, ponieważ szczyt liczby transakcji bazy danych w ciągu sekundy lub wielkość danych zwiększają się o rząd wielkości.

Używając wyznaczonych celów SLI i SLO, jak omówiliśmy to w rozdziale, można we wzroście znaleźć punkty, w których sensowne jest rozpoczęcie podziału danych na shardy funkcjonalne lub partycje danych. Skalowanie MySQL za pomocą shardingu omówimy dokładnie w rozdziale 11. W tym miejscu najważniejsze jest to, że te same cele SLI i SLO informujące o wydajności działania systemu mogą podpowiadać, kiedy nadszedł czas zainwestowania w MySQL, aby zarządzanie poszczególnymi klastrami pozostało możliwe w ramach celu SLO zapewniającego zachowanie wrażeń użytkownika.

Kup książkę Poleć książkę

Posiadanie rozwiązania przeznaczonego do obsługi wskaźników zarówno krótko-, jak i długoterminowych oraz śledzenie zmian trendu w użyteczny sposób ma bardzo duże znaczenie podczas monitorowania taktycznych wskaźników wydajności oraz długoterminowych trendów pokazujących, jaki infrastruktura bazy danych ma wpływ na działalność biznesową.

Podsumowanie

Podczas stosowania koncepcji inżynierii niezawodności w infrastrukturze monitorowania bazy danych ważne jest nieustanne usprawnianie i analizowanie wskaźników i celów. One nie muszą pozostać niezmienne po pierwszym zdefiniowaniu wartości SLI i SLO. Wraz z rozwojem firmy zyskujesz dokładniejszą wiedzę o wrażeniach użytkownika, którą należy wykorzystać do usprawnienia wskaźników SLI i SLO.

Należy świadomie wybierać wskaźniki i definiować ich cele oraz zawsze koncentrować się na przedstawianiu wrażeń użytkownika. Ponadto nie wolno koncentrować się tylko na wskaźnikach pokazujących, kiedy wydarzył się incydent, lecz poświęcić nieco czasu na monitorowanie aspektów, które mogą pomóc w *uniknięciu* incydentów. To jest rodzaj działalności proaktywnej, mającej na celu zabezpieczanie wrażeń użytkowników.

Zalecamy wcześniejsze zdefiniowanie celów na trzech obszarach: opóźnienie, dostępność i błędy. Te trzy aspekty mogą stanowić doskonały sposób sygnalizacji, czy klienci są zadowoleni. Ponadto należy stosować proaktywne monitorowanie w zakresie wzrostu liczby połączeń, ilości pamięci masowej, ilości dyskowych operacji wejścia-wyjścia oraz opóźnienia.

Mamy nadzieję, że materiał przedstawiony w tym rozdziale pomoże Ci w zastosowaniu inżynierii niezawodności do monitorowania MySQL podczas rozwoju firmy.

Kup książkę Poleć książkę

Kup książkę Poleć książkę


PROGRAM
PARTNERSKI
GRUPY HELION
1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ
Zmień swoją stronę WWW w działający bankomat!
Dowiedz się więcej i dołącz już dzisiaj!
http://program-partnerski.helion.pl
GRUPA
Helion

MySQL: skalowalne rozwiązanie do nowoczesnych zastosowań!

MySQL od lat jest najbardziej znaną i lubianą bazą danych typu open source. Wciąż spełnia oczekiwania użytkowników, staje się bowiem coraz bardziej zaawansowanym rozwiązaniem bazodanowym. Równocześnie rośnie złożoność tego oprogramowania, pojawiają się nowe funkcje i narzędzia. Ewoluuje też rola administratora bazy danych, podobnie jak zmienia się podejście do pracy zespołowej. Efektem tych zmian powinno być dostarczenie aplikacji o wysokiej wartości biznesowej, takiej, która będzie się skalowała wraz z organizacją. Warunkiem osiągnięcia tego celu jest jednak gruntowna znajomość MySQL.

Ten starannie zaktualizowany przewodnik pozwoli Ci poznać zaawansowane techniki pracy z serwerem MySQL: począwszy od tworzenia obiektów, poprzez projektowanie schematów, indeksów i zapytań, skończywszy na optymalizacji serwera, sprzętu i systemu operacyjnego. Opisano tu bezpieczne i praktyczne sposoby skalowania aplikacji za pomocą replikacji. Pokazano, jak można zapewnić równoważenie obciążenia i sprawić, że aplikacja będzie działała nawet w razie awarii. Omówiono najnowsze trendy pracy z serwerami MySQL (w tym bazy pracujące w chmurze), a także nowe funkcje i narzędzia. Nie zabrakło najlepszych praktyk w zakresie zapewniania bezpieczeństwa bazy, jej wydajności i stabilności. Dzięki tej pozycji zdobędziesz pełną wiedzę o tym, jak nowoczesne firmy używają MySQL na dużą skalę.

W książce między innymi:

- architektura MySQL i silniki pamięci masowej
- konfiguracja serwera a sprzęt
- replikacja MySQL i zapewnienie wysokiej dostępności aplikacji
- serwery MySQL w środowiskach zarządzanej chmury
- zaawansowane techniki optymalizacji MySQL
- automatyzacja zarządzania bazą danych

Silvia Botros jest architektem oprogramowania w Twilio. Brała udział w tworzeniu dużych aplikacji bazodanowych. Zajmowała się też projektowaniem i wdrażaniem magazynów danych w środowiskach produkcyjnych.

Jeremy Tinley jest starszym inżynierem w Etsy. Od ponad 20 lat zajmuje się bazami MySQL. Jest znany z umiejętności zapewniania swoim bazom danych dostępności, niezawodności i operacyjnej efektywności.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej!

ISBN 978-83-283-9294-6

9 788328 392946

Cena: 89,00 zł