

Równoważenie obciążenia chmur baz danych

Definicja chmury obliczeniowej o funkcjonalności DBaaS i modelu przetwarzania OLTP.

Dana jest publiczna chmura składająca się z N węzłów i oferująca zbiór interaktywnych usług obliczeniowych polegających na dostępie do umieszczonego w chmurze zbioru baz danych różnych najemców chmury. Funkcjonalność węzłów chmury obejmuje funkcje składowania i przetwarzania danych zgodnie z modelem danych wybranych systemów zarządzania bazami danych, np. relacyjnych lub NoSQL.

Bazy danych najemców chmury są dzielone na fragmenty danych (shards) - d_{ij} , gdzie i jest identyfikatorem bazy danych, a j identyfikatorem fragmentu danych. Fragmenty danych są alokowane na wybranych węzłach, $alloc(d_{ij}) = k$, gdzie k jest identyfikatorem węzła chmury, na którym alokowano fragment danych. Żądania dostępu do danych zawierają odwołania do danych. Oprogramowanie systemowe chmury (dispatcher) na podstawie identyfikatorów (wartości danych) wyznacza odpowiedni fragment danych d_{ij} i jego alokację k . Organizacja chmury obliczeniowej umożliwia składowanie na każdym z węzłów fragmentów danych pochodzących z baz danych wielu różnych najemców chmury (multitenant). Fragmenty danych mogą być replikowane. Te same fragmenty danych d_{ij} mogą być alokowane na zbiorze węzłów $R \subseteq N$. Średnia wartość $|R|/|N|$ dla wszystkich fragmentów danych będzie nazywana poziomem nasycenia chmury replikami danych.

Wydajność danego k -tego węzła chmury jest wyrażona za pomocą dwóch parametrów: mocy przetwarzania pp_k i potencjału równoległego przetwarzania pc_k . Moc przetwarzania pojedynczej usługi dostępu do danych jest wyrażona za pomocą maksymalnej liczby elementarnych jednostek pracy przetwarzanych w jednostce czasu. Potencjał równoległego przetwarzania jest wyrażony za pomocą maksymalnej liczby współbieżnych wątków, które mogą być obsługiwane bez opóźnień, które są konsekwencją niedostępności zasobów. Przetwarzane przez dany węzeł zadania są obsługiwane w trybie współdzielenia procesora. Obciążenie danego węzła obliczeniowego i w danej jednostce czasu t_j jest równe liczbie obsługiwanych elementarnych jednostek pracy we wszystkich współbieżnie obsługiwanych wątkach. Obciążenie danego węzła liczbą współbieżnych wątków większą od jego potencjału równoległego przetwarzania skutkuje pojawieniem się opóźnień, których wielkość wynosi dla każdego z równolegle przetwarzanych zadań wynosi: $((liczba\ wątków/pc_j)-1)*jednostka\ pracy/pp_k$.

Dodatkowo węzły charakteryzują się określoną pojemnością pamięci masowych dc_k przewidzianą do składowania baz danych.

Definicja strumienia żądań wykonania usług dostępu do danych.

Użytkownicy chmury obliczeniowej kierują do niej żądania wykonania usług dostępu do danych $r_{ts,i}$, gdzie ts jest znacznikiem czasowym momentu przedłożenia żądania, a i jest wyznaczonym systemowo identyfikatorem fragmentu zawierającego dane do przetworzenia.

Żądania dotyczą wykonania zadań obliczeniowych o rozmiarze określonym za pomocą liczby elementarnych jednostek pracy $size(r_{ts,i}) = s$. Czas odpowiedzi t_r systemu na żądanie wykonania pojedynczego wywołania usługi obliczeniowej jest sumą czasu wykonania zadania t_s , który jest ilorazem wielkości zadania i mocy obliczeniowej węzła, na którym jest wykonywany (s/pp_k) oraz czasu opóźnienia t_l , który jest efektem obciążenia danego węzła obsługą zadań w liczbie przekraczającej potencjał równoległego przetwarzania pc . Rozmiary czasów wykonania zadań są losowe i zgodne z rozkładem wykładniczym. Średni rozmiar zadania jest równy s_{avg} , a wariancja rozkładu rozmiarów zadań jest równa tsv .

Uporządkowany przez znaczniki czasowe zbiór wszystkich żądań wykonania dostępu do danego fragmentu danych d_i tworzy podstrumień fragmentu ss_i (*shard stream*). Suma podstrumieni adresowanych do danego węzła obliczeniowego k tworzy podstrumień węzła ns_k (*node stream*). W wypadku replikacji danych podstrumienie węzłów są formowane przez stosowane algorytmy wyboru repliki, których celem jest zagwarantowanie wymaganego poziomu spójności, dostępności i minimalizacji czasów odpowiedzi. Suma wszystkich podstrumieni fragmentów ns_k tworzy całkowity strumień żądań skierowany do chmury obliczeniowej CS (*cloud stream*). Częstość pojawiania się kolejnych zadań w podstrumieniach jest zmienna. Częstość pojawiania się kolejnych żądań w poszczególnych podstrumieniach zmienia się losowo i jest zgodna z rozkładem wykładniczym. Sumaryczny poziom obciążenia chmury w danej jednostce czasu $TL(t_j)$, jest sumą obciążeń wszystkich jej węzłów obliczeniowych $TL_i(t_j)$. Średni poziom obciążenia chmury przez strumień całkowity wynosi TL_{avg} .

Przebiegi obciążenia w poszczególnych podstrumieniach fragmentów ss_i charakteryzują się zadaniem poziomem skorelowania. Strumienie fragmentów ss_i tworzą K grup podstrumieni, o podobnej charakterystyce czasowej zmienności przebiegów obciążenia. Poziom skorelowania przebiegów obciążenia wewnątrz grup jest bardzo wysoki. Wartość średnia współczynnika korelacji jest bliska jedności. Z kolei poziom skorelowania różnych grup jest znacznie mniejszy. Średni współczynnik korelacji między podstrumieniami z różnych grup wynosi cc .

Model implementacji generatora strumieni żądań

Przebiegi zmienności intensywności podstrumieni ss_i są opisane liczbą żądań dostępu do bazy danych przedłożonych w jednostce czasu. Zmienne losowe reprezentujące zmienność tych przebiegów poszczególnych podstrumieni tworzą wektor: $\langle X_1, X_2, \dots, X_n \rangle$. Implementacja tego wektora wymaga utworzenia n niezależnych generatorów $\langle g_1, g_2, \dots, g_n \rangle$ generujących wartości o rozkładzie wykładniczym, różniących się wartościami średnimi i odchyłkami standardowymi. Wartości średnie i odchyłki dla intensywności poszczególnych podstrumieni są wyznaczane losowo (rozkład normalny albo wykładniczy). Suma wartości średnich powinna być mniejsza od sumarycznej mocy obliczeniowej chmury. Korzystając z

generatorów należy wygenerować n wektorów wartości intensywności poszczególnych strumieni w poszczególnych jednostkach danych.

Generatory	Kolejne jednostki czasu					
	1	2	3	4	...	m
X_1	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$		$V_{1,m}$
X_2	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$		$V_{2,m}$
...						
X_n	$V_{n,1}$	$V_{n,2}$	$V_{n,3}$	$V_{n,4}$		$V_{n,m}$

Następnie należy dokonać transformacji powyższych wektorów w celu skorelowania powyższych wektorów dla zadanego współczynnika korelacji. Dla rozkładów normalnych stosuje się w tym celu rozkład Choleskiego.

Generatory	Kolejne jednostki czasu					
	1	2	3	4	...	m
X_1	$V'_{1,1}$	$V'_{1,2}$	$V'_{1,3}$	$V'_{1,4}$		$V'_{1,m}$
X_2	$V'_{2,1}$	$V'_{2,2}$	$V'_{2,3}$	$V'_{2,4}$		$V'_{2,m}$
...						
X_n	$V'_{n,1}$	$V'_{n,2}$	$V'_{n,3}$	$V'_{n,4}$		$V'_{n,m}$

Realizacja eksperymentów

Eksperyment 1

Celem pierwszego eksperymentu jest ustalenie zależności między poziomem nasycenia chmury usługami dostępu do danych (równym liczbie replik danych) i poziomem obciążenia chmury, a sumarycznym i średnim czasem opóźnienia w dostępie do usług.

Dany jest zbiór F fragmentów danych rozrzuconych na N węzłach chmury. Każdy z fragmentów danych występuje w liczbie R replik. Na pojedynczym węźle mieści się M fragmentów, gdzie: $M=(F \cdot R)/N$. Moce obliczeniowe i potencjał zrównoleglania wszystkich węzłów są sobie równe. Średni poziom obciążenia chmury jest mniejszy lub równy od potencjału obliczeniowego chmury. Współczynniki korelacji między wszystkimi podstrumieniami fragmentów są sobie równe (nie ma wyróżnionych grup podstrumieni). Zakładamy nieograniczoną pojemność węzłów. Każdy z nich zmieści wszystkie ulokowane na nim fragmenty.

Należy zmierzyć czasy opóźnień w obsłudze żądań użytkowników oraz średnie obciążenie węzłów chmury dla różnych średnich poziomów obciążenia chmury, dla różnej liczby replik R i dla różnych współczynników korelacji.

Eksperyment 2

Celem drugiego eksperymentu jest ustalenie wpływu alokacji fragmentów danych o odmiennych (nieskorelowanych) przebiegach na tych samych węzłach obliczeniowych, na zmniejszenie czasów opóźnień w realizacji usług.

Dany jest zbiór F fragmentów danych rozrzuconych na N węzłach chmury. Każdy z fragmentów danych występuje w liczbie R replik. Na pojedynczym węźle mieści się M fragmentów, gdzie: $M=(F*R)/N$. Moce obliczeniowe i potencjał zrównoleglania wszystkich węzłów są sobie równe. Średni poziom obciążenia chmury jest mniejszy lub równy od potencjału obliczeniowego chmury. Podstrumienie fragmentów tworzą K grup skorelowanych wewnętrzną zmiennością przebiegów obciążenia. Współczynniki korelacji przebiegów obciążenia podstrumieni fragmentów wewnątrz grup są bliskie jedności. Współczynniki korelacji przebiegów obciążenia między grupami są konfigurowalne $(-\min;1)$. Dla poszczególnych współczynników korelacji należy wyznaczyć następujące parametry:

- Globalny parametr:
 $\text{stddev}(\text{CS}=\text{sum}(ns_i)=\text{sum}(ss_j)) / \text{sum}(\text{stddev}(ss_j))$, dla wszystkich ss_j ,
pokazuje potencjał chmury do zmniejszania opóźnień w wyniku właściwej alokacji fragmentów. Wartości parametru powinny należeć do przedziału: $<0; N>$ - im mniejszy tym lepszy.
- Wartość średnia lokalnych dla poszczególnych węzłów parametrów:
 $\text{stddev}(ns_i) / \text{sum}(\text{stddev}(ss'_j))$, dla wszystkich ss_j tworzących podstrumień ns_i , odzwierciedla jakość alokacji fragmentów. Dla dobrej alokacji wartości współczynników lokalnych powinny być bliskie parametrowi globalnemu.

Należy zmierzyć czasy opóźnień w obsłudze żądań użytkowników oraz średnie obciążenie węzłów chmury dla różnych średnich poziomów obciążenia chmury i dla różnych współczynników korelacji w trzech następujących przypadkach:

1. Podstrumienie węzłów ns_i zawierają żądania dostępu do fragmentów danych należących do tych samych grup. Odpowiada to sytuacji gdy, na tych samych węzłach są alokowane fragmenty o tej samej charakterystyce czasowej obciążenia.
2. Podstrumienie węzłów ns_i zawierają żądania dostępu do losowo wybranych fragmentów danych.
3. Podstrumienie węzłów ns_i zawierają żądania dostępu do fragmentów danych należących do różnych grup. Odpowiada to sytuacji gdy, inteligentna warstwa zarządzania alokuje na tych samych węzłach fragmenty o różnej charakterystyce czasowej obciążenia.

Przydatne będzie zestawienie wyników pierwszego i drugiego eksperymentu, żeby ustalić dla jakiego poziomu nasycenia usługami uzyskujemy czasy opóźnień podobne do tych z inteligentną alokacją.